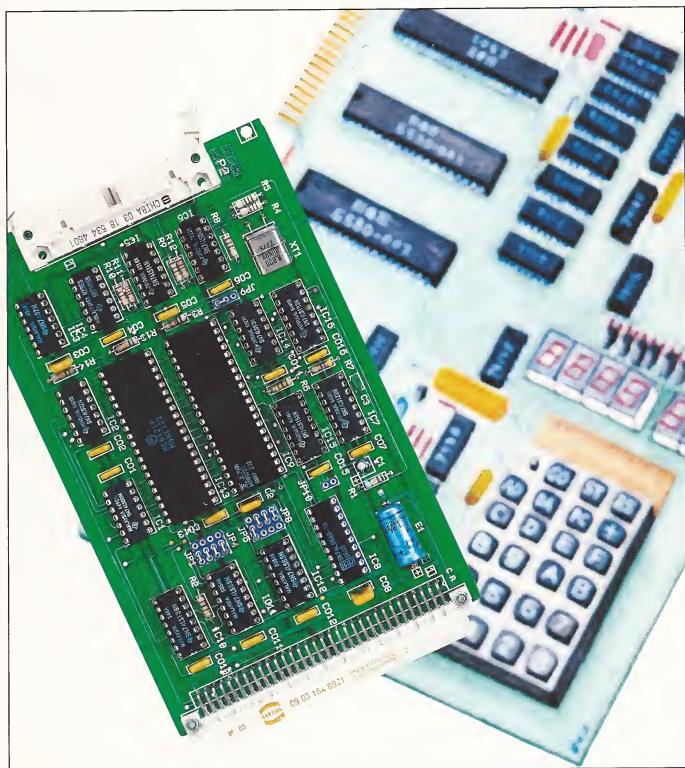


Twaalfde jaargang nr. 2 april 1988



### INFORMATIE.

De 6502 Kenner is een uitgave van de KLM Gebruikersclub Nederland. Deze vereniging is volledig onafhankelijk, is statutair opgericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Het doel van de vereniging is het bevorderen van de kennisuitwisseling tussen gebruikers van computers die zijn opgebouwd rond een microprocessor uit de 6500-familie. Voorbeelden hiervan zijn onder andere: Elektuur EC-65, Commodore 64, Apple ][, Elektuur Junior, Atari 600 en 800.

De eerder genoemde kennisuitwisseling komt onder andere tot stand door 6 maal per jaar de 6502 Kenner te publiceren, door de organisatie van landelijke bijeenkomsten voor de leden, het instandhouden van een softwarebibliotheek op cassette, floppy disk en papier en het beschikbaar stellen van een Bulletin Board.

### **Landelijke bijeenkomsten:**

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exacte plaats en datum worden steeds in de 6502 Kenner bekend gemaakt in de rubriek Uitnodiging.

### **Bulletin Board:**

Voor het uitwisselen van mededelingen, het stellen en beantwoorden van vragen en de verspreiding van software wordt er door de vereniging een Bulletin Board beschikbaar gesteld. Dit Bulletin Board valt onder de verantwoordelijkheid van één van de bestuursleden en wordt bediend door een zgn. Sysop.

### **Software Bibliotheek:**

Voor het beheer van de Software Bibliotheek streeft het bestuur er naar zgn. Software Coördinatoren te benoemen. Hierbij wordt gedacht aan een drietal coördinatoren; één voor EC-65(K) en Junior met OHIO OS-65D, één voor DOS-65 en één voor diverse andere systemen zoals onder andere Elektuur Junior.

### **Het Bestuur:**

Het bestuur van de vereniging wordt gevormd door een dagelijks bestuur bestaande uit een voorzitter, een secretaris en een

pennigmeester en een viertal gewone leden.

### **Voorzitter:**

Rinus Vleesch Dubois  
Florence Nighthingalestraat 212  
2037 NG HAARLEM  
Telefoon 023-330993

### **Secretaris:**

Gert Klein  
Diedenweg 119  
6706 CM WAGENINGEN  
Telefoon 08370-23646

### **Penningmeester:**

John van Sprang  
Tulp 71  
2925 EW KRIMPEN A/D IJSSEL

### **Leden:**

Adri Hankel  
Willem Kloosstraat 32  
7606 BB ALMELO  
Telefoon 05490-51151

Erwin Visschedijk  
Dillelaan 11  
7641 CX WIERDEN  
Telefoon: 05496-76764

Gert van Opbroek  
Bateweg 60  
2481 AN WOUBRUGGE  
Telefoon 01729-8636

Nico de Vries  
Mari Andriessenrade 49  
2907 MA CAPPELLE A/D IJSSEL  
Telefoon 010-4517154

### **Ereleden:**

Naast het bestuur zijn er een aantal ereleden, die zich in het verleden bijzonder verdienstelijk voor de club hebben gemaakt:

### **Erevoorzitter:**

Siep de Vries

### **Ereleden:**

Mevr. H. de Vries van der Winden  
Anton Mueller

=====

### De 6502 Kenner:

De 6502 Kenner wordt bij verschijnen gratis toegezonden aan alle leden van de KIM Gebruikersclub Nederland. De kopij voor het blad dient bij voorkeur van de leden afkomstig te zijn. Alle kopij wordt door de redactie op bruikbaarheid en publicatiewaarde beoordeeld. Deze twee criteria, in samenhang met de actualiteit, bepalen of en zo ja wanneer het stuk gepubliceerd wordt. De redactie streeft er naar de kopij zoveel mogelijk in zijn oorspronkelijke vorm te plaatsen, Nederlandstalige kopij wordt daarom in principe niet naar een andere taal vertaald. De redactie streeft er naar een Nederlandstalig blad te maken doch het staat de auteur vrij een artikel geheel of gedeeltelijk in een andere taal te schrijven.

Helaas kan de redactie, noch het bestuur, enige aansprakelijkheid aanvaarden voor de toepassing(en) van de gepubliceerde kopij.

### Verschijningsdata:

De 6502 Kenner verschijnt op de derde zaterdag van de maanden februari, april, juni, augustus, oktober en december.

### Redactie.

De redactie wordt gevormd door:

De bestuursleden.

Redactieadres:

Gert van Opbroek  
Bateweg 60  
2481 AN Woubrugge

### INHOUDSOPGAVE

#### Vereniging:

Informatie .....	2
Van de voorzitter .....	5
DOS-65 Coördinator .....	5

Uitnodiging clubbijeenkomst .....	6
-----------------------------------	---

#### Algemeen:

Redactioneel .....	4
Vragenrubriek .....	6
Computer Graphics .....	29
Bespiegelingen 65(C)02-'88++ .....	45
Reactie .....	47
Vragenrubriek .....	48
Gevraagd .....	48

#### DOS-65:

Vertaalprogramma van printcodes .....	7
De Viditel Karaktergenerator (2) .....	35

#### Hardware:

DOS-65K .....	41
---------------	----

#### Markt-Info

De video-controller CRT 9028/9128 .....	31
---	----

#### Software/Talen:

Simple driver for Thomson EF9367 .....	17
Datum --> Weekdag Conversie (F4th) .....	49

### REDACTIONEEL.

Tegen de tijd dat u dit leest, is het ongeveer een maand geleden dat ik dit geschreven heb. Dat betekent dat zaken die nog actueel zijn, dan al weer enigzins in het vergeethoekje geraakt zijn. Toch denk ik dat het zinvol is een paar zaken uit het recente verleden te bekijken.

In de eerste plaats is er in februari nummer 54 van de 6502 Kenner uitgekomen. Het was het eerste nummer dat volledig door mij samengesteld en geplakt is. Wel, dat was te zien. Enkele leden dachten dat ze al dronken waren toen ze met het blad begonnen te lezen. Dat was echter niet zo, de teksten en tekeningen waren niet erg strak. Ik denk dat het goed is eens uit te leggen hoe dat gekomen is.

Artikelen werden per kolom afgedrukt op een letterwielprinter. Deze stroken zijn toen op een vel A4 geplakt, samen met de bijbehorende tekeningen. Verder zijn er op dat vel de kopjes geplakt waarna met wrijffletters de pagina-nummering aangebracht werd. Hierna zijn de vellen A4 twee aan twee aaneelkaar geplakt en gekopieerd op een vel A3. Deze fungeerde als master voor de drukker. Nu heeft bij het fotokopieren het origineel niet voldoende vlak gelegen. Dat lag aan het feit dat bij het plakken te veel lijm gebruikt is en dat de stukken rondom vastgeliemd waren (het zou tenslotte zonde zijn als er kopij weg zou waaien).

Bij deze 6502 Kenner heb ik voor het plakken veel minder lijm gebruikt en de stukken ook slechts op een paar hoekjes vastgeplakt. Bovendien heb ik de beide kolommen voor een pagina rechtstreeks op het juiste formaat afgedrukt zodat meestal alleen de kopjes geplakt hoefden te worden. Originale listings en figuren zijn ook met zo weinig mogelijk lijm geplakt. Wel ben ik van plan van een dergelijke pagina nog een fotokopie op A4 te maken om deze bij de drukker in te leveren. Ik hoop dat hierdoor het uiterlijk van het blad nog iets verbetert.

Een tweede punt waarover ik iets moet schrijven is het volgende. Ik wil graag een blad uitgeven met een zo divers mogelijke inhoud. Ik ben echter niet in staat in mijn eentje over van alles en nog wat te schrijven en het maken van programma's voor EC-65 en DOS-65 kan ik helemaal niet omdat ik niet een dergelijk systeem tot mijn beschikking heb (ik heb daar overigens ook geen belang bij). Dat betekent dat ik voor machine-afhankelijke zaken

volledig afhankelijk ben van u! Ik roep dan ook iedereen op kopij in te zenden waarbij ik voornamelijk denk aan programmatuur en hardware voor de EC-65(K) omdat ik toch wat gemakkelijker toegang heb tot DOS-65 spullen. Het feit dat er in dit blad geen EC-65 voorkomt, wordt enkel en alleen veroorzaakt door het feit dat er geen kopij beschikbaar is. Wel weet ik dat de software coördinatoren voor dit systeem begonnen zijn aan een inventarisatie van software en operating systemen en ik hoop dat er spoedig weer een zeer actieve EC-65(K) groep binnen de club is. Kortom, hebt u voor uw systeem iets ontwikkeld, laat het mij dan op zijn minst even weten zodat we er gezamenlijk misschien een artikelje over kunnen schrijven. Een voorbeeld van een dergelijke samenwerking vindt u in het verhaaltje van de heer P. de Vries over de CRT9128.

Dan een volgend punt. Zoals in nummer 54 verteld is, bezit de club een Bulletin Board. Dit systeem draait momenteel heel aardig en wordt door een aantal mensen (waaronder ikzelf) zeer frequent bezocht. Het blijkt dat er via dit systeem snel contact gelegd kan worden tussen mensen met gelijke interesses of problemen. Bovendien blijkt dat auteurs van software via het Bulletin Board meldingen doen van aanvullingen en verbeteringen van hun producten.

Ik wil dit systeem een centrale plaats geven in de verspreiding van software. Dat betekent dat elk in de 6502 Kenner gepubliceerd programma bij verschijnen van de 6502 Kenner in principe ook op het Bulletin Board staat waar het dan door leden kosteloos afgehaald kan worden. Voor de mensen die geen modem hebben, kan men de software ook via de softwarecoördinator krijgen en eventueel via de redactie.

Hoe de preciese procedure wordt voor de verspreiding via tape of diskette, zal ik in de volgende 6502 Kenner beschrijven. Uitgangspunt wordt hierbij wel dat materialen tegen kostprijs verspreid worden.

Tenslotte heb ik nog een oproep. Ik denk dat het goed is, dat de tekst in de 6502 Kenner zo hier en daar verlicht wordt met een plaatje. Het lijkt mij wel leuk in het blad een soort strip te hebben met als hoofdpersoon een chipje (sjippie?). Nu komt echter het probleem dat ik niet kan tekenen. Wie helpt?? Andere plaatjes zijn ook welkom.

Verder veel hobbygenot aan 6502 Kenner 55. Uw redacteur, Gert van Opbroek

### VAN DE VOORZITTER

Was u ook van de partij, op de laatste bijeenkomst te Geldrop? Neen dan heeft u een kans gemist om gezellig met uw mede systeem-bouwers van gedachten te wisselen of mogelijk een probleem te helpen oplossen. De volgende bijeenkomst krijgt u deze kans wederom, dus komt u ook, en neem als mogelijk uw systeem mee! Hoe vindt u overigens de inhoud en kwaliteit van de laatste 6502-kenners? Onze nieuwe redakteur Gert van Opbroek heeft dat bijna in z'n eentje geklaard, knap he! Maar toch zou het bestuur graag een beroep op uw medewerking willen doen, dus biedt onze redakteur uw hulp, door kopij in te sturen (u kunt het best) of door u op te geven als redakteur medewerker. Voor dit laatste kunt u contact opnemen met onze redakteur. Heeft u al contact gehad met het bulletin-board? Ook daar kunt u uw kopij of programmatuur naar toe sturen. Overigens over communicatie gesproken, wist u dat dit een van de snelst groeiende technieken is, in 1989 wordt onze nationale PTT geprivatiseerd, dit zal mede tot gevolg hebben dat de concurrentie flink zal toenemen, men raakt namelijk de monopoly positie kwijt. Ik denk dat dit de ontwaakding sterk ten goede zal komen en daardoor mogelijk ook de prijzen die de consument moet betalen. Stel u eens voor dat de kosten voor modem gebruik door iedereen betaalbaar zou zijn, dan opent dit toch ongekende mogelijkheden voor alle computer hobbyisten. Deze lage kosten zullen ondermeer bereikt kunnen worden door verbetering van het kabelnetwerk, denk maar aan glasvezel-kabels of misschien zelfs door kabels met super geleiding, in ieder geval zal de baudrate (dat is de snelheid waarmee data wordt overgestuurd) flink omhoog moeten worden geschroefd. Uit een en ander blijkt, dat als men zich voor de techniek interesseert, men nog veel zal beleven. Ook het bestuur van uw club stuit regelmatig op dit onderwerp. Tijdens de bestuursvergaderingen wordt er veel gebrainstormd over de toekomst, het valt niet mee om steeds weer wat nieuws te bedenken voor onze leden, maar anderszijds is het best een uitdaging. Krijgt u toeval of bewust ook een goed idee wat bruikbaar is voor onze club, laat mij dat dan even weten. Ik wens u veel leesplezier met deze uitgave van onze 6502 kenner, en zie u graag terug op de komende bijeenkomst.

Rinus Vleesch Dubois.

### DOS65 COÖRDINATOR

Er heeft zich een zeer enthousiaste DOS65 gebruiker gemeld als nieuwe coördinator. Vanaf nu kan iedere DOS65 gebruiker terecht bij:

Jan Derksen  
Cp. Soeteliefstraat 41  
1785 CC Den Helder  
tel. 02230-35002

Aangezien Jan momenteel nog in militaire dienst is, is hij door de week telefonisch niet te bereiken. U kunt natuurlijk altijd schrijven.

Jan zal zich bezighouden met:

- de distributie van hardware en software van DOS65 artikelen,
- software en hardware ondersteuning van mede DOS65 gebruikers,
- het onderhouden van contacten met DOS65 gebruikers.

De DOS65 coördinator onderhoudt contacten met alle DOS65 gebruikers. Dit contact komt van beide zijden. Als u van plan bent een min of meer groot project te beginnen, bv. het ontwerpen van een nieuw stuk hardware of het schrijven van een brok software, laat dat de DOS65 coördinator dan ook weten. Misschien zijn anderen er al mee bezig. U kunt dan eventueel gaan samenwerken. Zo wordt voorkomen dat twee keer het wiel wordt uitgevonden en dat er op meerdere plaatsen veel energie in dezelfde zaken wordt gestoken.

Jan weet zeer veel van het DOS65 systeem, zodat ik hem dit werk zeer goed toevertrouw. Ik wens hem veel succes toe bij dit dankbare werk.

Ik hoop dat ik nu meer tijd zal overhouden voor andere DOS65 dingen die ik ook leuk vindt.

Helaas misschien voor DOS65 heb ik ook nog zeer veel andere hobby's waar ik ook graag wat aan wil doen. De reden waarom ik vroeger ooit eens een Junior heb gekocht was mijn ideaal om een trein-emplacement te automatiseren. Door ruimtegebrek is dat er nooit erg van gekomen. Nu heb ik wel de ruimte en begint mijn bloed sneller te stromen bij de gedachte aan computer gestuurde snelheidsregelingen, blokbeveiligingen, dienstregelingen enz.

Erwin Visschedijk.

### UITNODIGING CLUBBIJEENKOMST

Datum: 28 mei 1988 (een week later dan normaal i.v.m. Pinksteren).  
Lokatie: Kerkgebouw "DE BRON"  
Hobbemalaan 1  
2923 XD Krimpen a/d IJssel  
tel: 01807-16287

Entreprijs: fl. 10,--

### **Routebeschrijving**

#### **AUTO:**

Komende uit de richting Utrecht. Volg de autoweg Utrecht-Rotterdam-Dordrecht tot de afslag Cappelle a/d IJssel voor de v. Brienoordbrug. Afslag rechtsaf richting Cappelle a/d IJssel voert linksaf onder de weg naar de v. Brienoord door. Houdt richting Cappelle a/d IJssel aan tot de rotonde Cappelse plein, waarop (zie richtingaanwijzers) rechtsaf richting Schoonhoven en Krimpen a/d IJssel. Deze weg voert over de Algrabrug. Een stukje verder ontmoet u stoplichten. Ga hier linksaf de Nieuwe Tiendweg op en volg deze tot de volgende rotonde. Ga hier rechtsaf de Burg. Aalberslaan op, dan de eerste zijstraat rechts, de Jan van Goyenstraat in, dan de eerste zijstraat rechts, de Hobbemalaan, in tot aan het eind. Daar kunt u parkeren. Aan de rechterhand het vlakke kerkgebouw (zie pijl op kaartje bij no. 30).

Komende uit de richting Amsterdam. Volg de A4 Amsterdam-Rotterdam. Knooppunt Leidschendam passeren. Verder richting Dord-

recht via v. Brienoordbrug aanhouden. Voor de v. Brienoordbrug afslag Cappelle a/d IJssel nemen. Vervolgens als hierboven.

Komende uit de richting Rotterdam of Dordrecht. Richting Cappelle a/d IJssel aanhouden en verder als bij de eerste beschrijving.

#### **OPENBAAR VERVOER:**

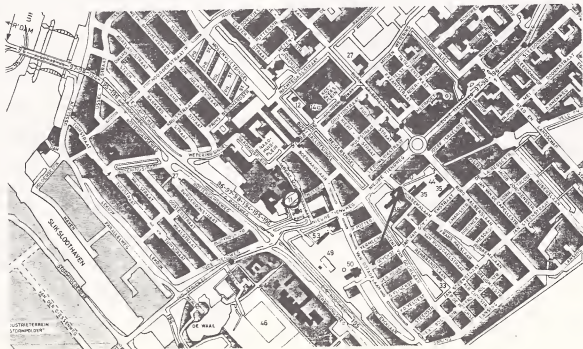
Per trein naar station Rotterdam-Alexanderpolder. Hiervandaan met de Metro naar de Cappelse brug en verder met de bus (o.a. lijn 98) naar Krimpen a/d IJssel. Uitstappen bij het busstation Krimpen a/d IJssel. Vanaf het busstation (nr. 11 op plattegrond) linksaf de Nieuwe Tiendweg op en aan de tweede straat rechts vindt u "De Bron".

#### **Programma:**

9:30 Zaal open met koffie  
10:15 Opening  
10:30 Voordracht van Ruud Uphoff:  
De kwaliteitszorg van software.  
11:30 Forum en markt  
12:00 Lunchpauze

Aansluitend het informele gedeelte bedoeld om kennis, ervaring Public Domain en eigen ontwikkelde software uit te wisselen met uw medeleiden. **BRENG DAAROM OOK UW EIGEN SYSTEEM MEE!** (En vergeet de snoeren niet.....)

17:00 Sluiting.



```

***** D O S - 6 5 V2.01 *****
*
*   Vertaalprogramma van printcodes in een EDITFILE naar   *
*   een PRINTFILE voor alle EPSON achtige printer.         *
*
*****

```

Door: H.A.J.Quast  
 Dekemastate 15  
 1275 CM Huizen N.H.  
 tel. 02152-54905

Sinds enige tijd ben ik in het bezit van een PHILIPS NMS-1431 matrix printer. Dit is een printer die bedoeld is als uitbreiding voor een MSX-computer. De karakterset van deze printer komt ongeveer overeen met de standaard karakterset van de Epson printers.

Buiten de gewone ascii-tekenen, die deze printer kan weergeven, bezit de printer ook de mogelijkheid om een hele reeks van bijzondere schrijftekenen weer te geven. Deze reeks van tekens begint met de ascii-waarde \$80 t/m \$FF en vervolgens van 1,40 t/m 1,5F. Als we de Epson en Philips karakterset boven de \$7F met elkaar vergelijken, blijkt er grote overeenkomst te zijn. Alleen de groep tekens boven de \$FF bestaan niet bij de Epson printers en de grafische tekens kloppen niet.

Mijn bedoeling was nu om al deze bijzondere karakters op te kunnen geven in de edit-file.

De volgende punten waarmee ik zat waren:

- In de edit-file kunnen geen tekens boven de \$7F worden ingevoerd, zodat het gebruik van b.v. tekens als een 'e' met puntjes of met het schuine streepje erop niet mogelijk is.
- Een oplossing voor dit probleem was een vereenvoudigde versie van het hier beschreven programma die met ingetypte hex-codes werkte. Maar het nadeel hiervan was dat de file door al de hex-codes slecht leesbaar werd.

Een volgend probleem was:

- Als ik een file had aangemaakt waarin de speciale printerhex-codes zaten, kon ik deze niet zomaar naar iemand anders toesturen die met een ander type printer werkte. Het leek mij nuttig om een programma te maken waarmee dit voor een aantal type printers wel mogelijk was.

#### Beschrijving van het programma "CONVPRINT"

---

In dit programma wordt een speciale file (Printer-Conversie-File) van disk in het geheugen in een tabel geladen. Deze Printer-Conversie-File (PCF.DAT) moet op de systeem-disk staan en met de editor worden aangemaakt.

De opbouw van de file moet als volgt zijn:

Het eerste karakter op een regel is het teken dat in de text-file staat als vervanging van een bepaald printersymbool. Vervolgens moet er een # teken staan waarna het programma de eerste twee daaropvolgende karakters leest en deze samenvoegt tot een hexadecimaal karakter. Hierachter mag commentaar gezet worden, b.v. wat voor een soort teken het is. Dit commentaar moet wel voorafgegaan worden door een (;) teken. Ook is het mogelijk om meerdere letterdefinities achter elkaar te zetten, gescheiden door 1 of meerdere spaties.

Regels met alleen maar commentaar zijn ook toegestaan mits deze wel voorafgegaan worden door het teken (;).

Voor de edit-vervangcodes kunnen alle ascii-karakters worden gebruikt behalve het teken ; en het teken #. Wanneer de routine gebruikt wordt die geschikt is voor het werken met de Philips printer NMS-1431 dan mag ook het cijfer 1 niet gebruikt worden.

Om het programma niet te traag te maken bij het opzoeken van de vervangingscode is de tabel op ascii-code gesorteerd.

De gegevens uit de file worden op de volgende manier behandeld: Het eerste karakter dan gelezen wordt als Editor vervang-code wordt gebruikt als index-waarde voor de tabel PCFTAB. Vervolgens leest het programma verder totdat deze de printer-code heeft samengesteld. Deze waarde wordt in de tabel PCFTAB gezet op de plaats aangewezen door de index.

### Aanmaken tekst-file.

In de EDIT-mode wordt op de normale manier de tekst ingetypt. Wanneer er nu een speciale printcode gebruikt moet worden dan gaat dat als volgt:

Op de plaats van de speciale code typt men <CTRL+V>R gevolgd door de ascii-vervangcode van de printcode.

Voorbeeld:

Er is maar <CTRL+V>Re<CTRL+V>Ren IC kapot.

Nadat de inputfile gesloten is wordt het programma "CONVPRINT" aangeroepen met:  
CONVPRINT <inputfile> <outputfile>

(b.v. CONVPRINT TEKST.DOC TEKST.PRI)

Het programma leest nu een kar. uit de inputfile en test of deze de waarde <CTRL+V>R heeft (hex \$12). Wordt deze code gevonden dan wordt de hexwaarde van het daarachterstaande karakter gebruikt om als index te dienen voor de tabel PCFTAB.

Het byte dat hiermee aangewezen wordt is de printercode van het ascii-vervangteken.

Als het controlteken (\$12) niet gevonden wordt, schrijft het programma de gelezen ascii-code direct door naar de outputfile.

Door de outputfile uit te printen met ">P Plist <optie> <filename>" of met " PRINT (voor printen vanuit de spooler)" wordt het gewenste resultaat verkregen.

Voor het werken met de printer NMS-1431 van PHILIPS is er een extra routine ingebouwd die het mogelijk maakt om printercode's die beginnen met 1,xx (waarbij xx een getal is tussen de \$40 en de \$5F) ook uit te printen. De code in de editfile moet dan als volgt zijn:

Voorbeeld:

Printen van drie maal het muzieknootje.

```
intypen          : <CTRL+V>R14E      <CTRL+V>R14E      <CTRL+V>R14E
                  ||||              ||||              ||||

in de input file  | | | | | | | | | | | | | | | |
staat de code    : $12,$31,$34,$45  $12,$31,$34,$45  $12,$31,$34,$45
                  | | | | | | | | | | | | | | | |
het programma    | | | | | | | | | | | | | | | |
vertaald dit in  : $01, $4E        $01, $4E        $01, $4E
```

Deze vertaalde code wordt in de outputfile weggeschreven.

Enkele opmerkingen:

Het programma leest tijdens het vertalen steeds een sector van de inputfile. Na de genoemde vertaalslag wordt de data naar de outputfile weggeschreven. Wanneer we nu een erg lange file moeten vertalen is het verstandig om de ene file op de systeemdisk te zetten en de andere file op de userdisk. Het voordeel hiervan is dat de koppen van de diskdrive zich minder behoeven te verplaatsen. Ze blijven op de plek wachten waar ze gestopt zijn totdat de volgende sector gelezen of beschreven moet worden. Zouden we dezelfde file alleen op de user/systeemdisk vertalen dan moet de kop van de diskdrive zich voortdurend tussen de lees en de schrijfsector verplaatsen.

Een tweede opmerking is de volgende:

Nadat de file geconverteerd is mag deze niet meer via de editor bewerkt worden. De reden hiervan is de volgende:  
Stel dat we de volgende code in de inputfile hebben gezet:

```
<CTRL+V>Re      deze code geeft op de printer een e met een
                  schuin streepje.
```

```
In de inputfile staat de ascii-code $12,$65
|
het programma vertaald dit naar ..... $82
```

Wanneer nu deze file opnieuw in de editor geladen wordt dat zal de code \$82 verwijderd worden omdat hij boven de hexwaarde \$7F is. Met als gevolg dat het gewenste karakter niet afgedrukt wordt.

Als derde wil ik nog even wijzen op het gebruik van de utiliteit LIST. Aangezien deze utiliteit alle karakters die niet tot de standaardascii-karakterset behoren wegfilterd is deze dus niet te gebruiken voor het uitprinten van de outputfile.

Wanneer er achter een printcode een ongeldige ascii-waarde staat dan stopt het programma met vertalen, sluit de outputfile en print op het scherm een foutmelding met daaronder de tekst vanaf de plaats waar de fout ontdekt is tot aan de eerstvolgende CR.

De procedure van werken kort samengevat:

- openen inputfile met: EDIT
- file met speciale char. aanmaken in de editor.
- inputfile sluiten.
- aanroepen van programma CONVPRINT.
- uitprinten van de outputfile. ( >P PLIST optie outputfile )  
( PRINT optie outputfile )
- eventuele wijzigingen aanbrengen in de inputfile.
- en opnieuw vertalen.

Uitwisselbaarheid van de edit-vervangcode.

Om de edit-vervangcode en de bijbehorende printercode tussen verschillende gebruikers onderling uitwisselbaar te maken, is het van belang dat de inhoud van de file PCF.DAT gestandaardiseerd wordt. Ik heb zelf een poging gedaan maar het lijkt mij nuttig als geïnteresseerde hierover ook hun mening geven zodat er misschien een standaardfile gemaakt kan worden.

Voorbeeld opbouw van de PCF.DAT file.

```

;
;
; Printer-code file voor de EPSON printers
;
;
f # 9F ; f gulden teken
? # A8 ; vraagteken op zijn kop
~ # F7 ; ongeveer teken
^ # EC ; oneindig
/ # F6 ; Deelteken
W # FB ; Wortel
;
2 # AB ; Breukteken 1/2
4 # AC ; Breukteken 1/4
+ # F1 ; plus/min teken
0 # ED ; lege verzameling
{ # EE ; element verzameling
} # EF ; doorsnede
= # F0 ; equivalent met
> # F2 ; >=
. # AF ; vele malen groter
< # F3 ; <=
, # AE ; vele malen kleiner
;

A # 84 ; puntjes op de a
a # A0 ; streepje op de a
E # 89 ; puntjes op de e
e # 82 ; streepje op de e
O # 99 ; puntjes op de o
o # A2 ; streepje op de o
i # 8B ; twee puntjes op de i
;
@ # E0 ; alfa
B # E1 ; Beta
G # E2 ; Gamma
d # EB ; Delta kleine letter
N # EE ; Epsilon
t # E9 ; theta
u # E6 ; mu teken
P # E3 ; Pi
S # E4 ; hoofdletter sigma
s # E5 ; kleineletter sigma
T # E7 ; Tau
F # E8 ; fi
h # EA ; ohm-teken
;
; Einde printer-code file
;

```

```

; ***** D O S - 6 5 V2.01 *****
;
; *
; * Vertaalprogramma van EDITFILE naar PRINTFILE
; * voor de EPSON printer.
; *
; * In dit programma wordt een speciale conversie-file
; * van disk in het geheugen gelezen. De inhoud van de file
; * wordt in een tabel opgeborgen. In de tabel staan de
; * vervaningscodes van de letters die in de tekst staan
; * achter de speciale code. Om het programma niet te traag
; * te maken bij het opzoeken van de vervaningscode is de
; * tabel op ascii-code gesorteerd.
; * De gang van zaken is als volgt:
; * Het speciale printerkarakter wordt gelezen uit de file.
; * Vervolgens kijkt het programma wat het volgende teken is.
; * De asciiwaarde van dit teken is een index voor de tabel.
; * Het nadeel is dat de tabel meestal langer is dan strikt
; * noodzakelijk maar het voordeel is dat het vergelijken
; * en opzoeken veel sneller gaat.
; *
; *****
;
;
; Start commando: CONVPRINT <inputfile> <outputfile>
;
; Voor info type: HELP CONVPRINT
;
;
; Pagina nul adressen
; -----
00A0      ORG      $00A0
00A0      FILEIN  RES      1      Inputfilenummer
00A1      FILEOUT RES      1      Outputfilenummer
00A2      PCFILE  RES      1      Conversiefilenummer
00A3      DATASAV RES      1      Saveadres data
00A4      BUFINDX RES      1      Conversiebuffer-index
;
;
0012      SUBLET  EQU      $12      Printcontrol-char
;
; Gebruikte subroutine's
; -----
;
;
C023      OUT      EQU      $C023      Print karakter op het scherm
C03B      PRINT   EQU      $C03B      Print een string op het scherm
C03E      ASCHEX   EQU      $C03E      Ascii naar hex
D03F      OPEN    EQU      $D03F      Open een input-file
D036      CREATE   EQU      $D036      Creer een output-file
D048      CLOSE   EQU      $D048      Close de openstaande-file
D000      SREAD   EQU      $D000      Single read uit file
D009      SWRITE  EQU      $D009      Single write in file
D0B7      ERMES   EQU      $D0B7      Error melding
;
;
;
; -----
; Hoofdprogramma
; -----
1000      ORG      $1000

```

```

1000 4C F511      ; CONV      JMP      CONVPRINT
1003 8C5CCD0      FCC      $C8,$C5,$CC,$D0 Zoekstring voor help utilitie
1007 4C4C4C      FCC      $4C,$4C,$4C      Dummy bytes
100A 0D          INFO      FCC      '\r'
100B 46756E6374      FCC      'Function      : PRINTER-code vertaalt routine (EPSON uitvoering)
1010 696F6E2020      \Eo\r'
1015 203A205052
101A 494E544552
101F 2D636F6465
1024 2076657274
1029 61616C726F
102E 7574696E65
1033 2028455053
1038 4F4E207569
103D 74766F6572
1042 696E672920
1047 1B6E0D
104A 2020202020      FCC      '      : Dit programma zet de asciivervang-code\r'
104F 2020202020
1054 203A204469
1059 742070726F
105E 6772616D6D
1063 61207A6574
1068 2064652061
106D 7363696976
1072 657276616E
1077 672D636F64
107C 650D
107E 2020202020      FCC      '      : met behulp van een tabel voor de speciale\r'
1083 2020202020
1088 203A206D65
108D 7420626568
1092 756C702076
1097 616E206565
109C 6E20746162
10A1 656C20766F
10A6 6F72206465
10AB 2073706563
10B0 69616C650D
10B5 2020202020      FCC      '      : printkarakters om naar de printer codes.\r'
10BA 2020202020
10BF 203A207072
10C4 696E746B61
10C9 72616B7465
10CE 7273206F6D
10D3 206E616172
10D8 2064652070
10DD 72696E7465
10E2 72636F6465
10E7 732E0D
10EA 2020202020      FCC      '      : Hierbij wordt een outputfile aangemaakt met\r'
10EF 2020202020
10F4 203A204869
10F9 657262696A
10FE 20776F7264
1103 742065656E
1108 206F757470
110D 757466696C
1112 652061616E

```

```

1117 67656D6161
111C 6B74206D65
1121 740D
1123 2020202020      FCC      '      : de ASCII-codes plus de vertaalde printcodes.\r'
1128 2020202020
112D 203A206465
1132 2041534349
1137 492D636F64
113C 657320706C
1141 7573206465
1146 2076657274
114B 61616C6465
1150 207072696E
1155 74636F6465
115A 732E0D
115D 53796E7461      FCC      'Syntax      : CONVPRINT <inputfile> <outputfile>\r'
1162 7820202020
1167 203A20434F
116C 4E56505249
1171 4E54203C69
1176 6E70757466
117B 696C653E20
1180 3C6F757470
1185 757466696C
118A 653E0D
118D 4E6F206F70      FCC      'No options :\r'
1192 74696F6E73
1197 203A0D
119A 2020202020      FCC      '      : Printcommando voor de outputfile :\r'
119F 2020202020
11A4 203A205072
11A9 696E74636F
11AE 6D6D616E64
11B3 6F20766F6F
11B8 7220646520
11BD 6F75747075
11C2 7466696C65
11C7 203A0D
11CA 2020202020      FCC      '      : >P PLIST [optie] outputfile\r\r',0
11CF 2020202020
11D4 203A203E50
11D9 20504C4953
11DE 54205B6F70
11E3 7469655D20
11E8 6F75747075
11ED 7466696C65
11F2 0D0D00

;
; Initialiseer de inputfile
;
11F5 A2 81      CONVPRINT LDX      #$81
11F7 20 3FD0      JSR      OPEN      Open inputfile
11FA 90 03      BCC      1.f
11FC 4C B012      JMP      ERROR1
11FF 86 A0      STX      FILEIN      Save inputfilenummer
;
; Initialiseer de outputfile
;
1201 A9 FA      LDA      #$FA      Next file-name
1203 A2 E1      LDX      #$E1      ASCII-file

```

```

1205 20 36D0      JSR     CREATE      Maak outputfile aan
1208 90 03        BCC     1.f
120A 4C B012      JMP     ERROR1
120D 86 A1        STX     FILEOUT      Store het outputfilennummer
;
; Laad de printer-conversie-file
;
120F A9 13        LDA     #PCFBUF>>8  High-byte filenaambuffer
1211 A0 42        LDY     #PCFBUF&255  Low-byte filenaambuffer
1213 A2 81        LDX     #$81
1215 20 3FD0      JSR     OPEN         Open conversie-file
1218 90 03        BCC     1.f
121A 4C B012      JMP     ERROR1
121D 86 A2        STX     PCFILE       Save het filennummer
;
; Lees de conversiefile
;
121F 20 1B13      LSPCF  JSR     RDFIL   Lees data uit conversiefile
1222 F0 06        BEQ     1.f           # teken gevonden
1224 90 28        BCC     ENDPCHF       eof gevonden
1226 85 A4        STA     BUFINDX       Save de bufferindex
1228 D0 F5        BNE     LSPCF
122A 20 1B13      1      JSR     RDFIL   Lees data uit conversiefile
122D 90 1F        BCC     ENDPCHF       eof gevonden
122F 20 3ECO      JSR     ASCHEX        Ascii naar Hex conversie
1232 B0 3B        BCS     ASCERR        Stop met lezen
1234 0A          ASLA
1235 0A          ASLA
1236 0A          ASLA
1237 0A          ASLA
1238 85 A3        STA     DATASAV       Save data
123A 20 1B13      JSR     RDFIL   Lees data uit conversiefile
123D 90 0F        BCC     ENDPCHF       eof gevonden
123F 20 3ECO      JSR     ASCHEX        Ascii naar Hex
1242 B0 2B        BCS     ASCERR        Stop met lezen
1244 05 A3        ORA     DATASAV       Maak hex-byte
1246 A6 A4        LDX     BUFINDX
1248 9D 4C13      STA     PCFTAB,X      Save de data
124B 4C 1F12      JMP     LSPCF        Lees volgende code
;
; Sluit de printercode file
;
124E A6 A2        ENDPCHF LDX     PCFILE  Printercode filennummer
1250 20 48D0      JSR     CLOSE        Sluit de file
;
; Lees de inputfile en vertaal deze
;
1253 A6 A0        READ  LDX     FILEIN   Laad het inputfilennummer
1255 20 00D0      JSR     SREAD        Single read uit file
1258 B0 4B        BCS     RDEND        Test op end of file
125A C9 12        CMP     #SUBLET      Vergelijk met printercontrol-char.
125C D0 09        BNE     NOSUBL       jump als het ascii data is
125E 20 00D0      JSR     SREAD        Single read uit file
1261 B0 42        BCS     RDEND        Test op end of file
1263 AA          TAX
1264 BD 4C13      LDA     PCFTAB,X      Haal code uit de tabel
1267 A6 A1        NOSUBL LDX     FILEOUT  Laad het outputfilennummer
1269 20 09D0      JSR     SWRITE       schrijf ascidata in file
126C 4C 5312      JMP     READ         Haal volgend kar. uit de file
;

```

; Error melding voor een niet ascii-karakter

;

```
126F 20 3BC0  ASCERR JSR    PRINT
1272 0D1B692041  FCC      '\r\Ei ASCII-ERROR in printer-conversie file \En\r',0
1277 534349492D
127C 4552524F52
1281 20696E2070
1286 72696E7465
128B 722D636F6E
1290 7665727369
1295 652066696C
129A 65201B6E0D
129F 00
```

;

; Close de files

;

```
12A0 A6 A2  PCFCLO LDX    PCFILE      Printercode filennummer
12A2 20 48D0      JSR    CLOSE      Sluit de file
12A5 A6 A0  RDEND  LDX    FILEIN     Laad het inputfilennummer
12A7 20 48D0      JSR    CLOSE      Sluit de file
12AA A6 A1      LDX    FILEOUT     Laad het outputfilennummer
12AC 20 48D0      JSR    CLOSE      Sluit de file
12AF 60      RTS      terug naar DOS-65
```

;

```
12B0 4C B7D0  ERROR1 JMP     ERMES      print error
```

;

; Als er een code gelezen wordt welke niet omgezet

; kan worden naar een ascii-nibble volgt er een

; foutmelding.

;

```
12B3 20 3BC0  CERROR JSR    PRINT      Geef printercode error
12B6 0D1B692041  FCC      '\r\Ei ASCII-ERROR in printercode \En\r\r'
12BB 534349492D
12C0 4552524F52
12C5 20696E2070
12CA 72696E7465
12CF 72636F6465
12D4 201B6E0D0D
12D9 6E65787420  FCC      'next characters follows after the error\r\r'
12DE 6368617261
12E3 6374657273
12E8 20666F6C6C
12ED 6F77732061
12F2 6674657220
12F7 7468652065
12FC 72726F720D
1301 0D
```

```
1302 3D3D3E2000  FCC      '==> ',0
1307 A6 A0      LDX    FILEIN     Laad het inputfilennummer
1309 20 00D0  LOOPERR JSR    SREAD      Single read uit file
130C 20 23C0      JSR    OUT      Print het kar. op het scherm
130F C9 0D      CMP     #$0D      Test op einde van de tekstregel
1311 D0 F6      BNE     LOOPERR
1313 A9 0A      LDA     #$0A
1315 20 23C0      JSR    OUT
1318 4C A512      JMP     RDEND
```

;

;

; \*\*\*\*\*

; Routine: RDFIL

```
;
; Lees byte uit de file en test op een REMARK, SPATIE
; en # teken.
; Bij een REMARK wordt er doorgelezen tot het einde van
; de regel, een spatie wordt overgeslagen.
; Wanneer het # teken gelezen wordt verlaat het programma
; de routine met de Z-vlag als status voor het vinden
; van het # teken. Z-vlag = 1 teken gevonden.
; Wanneer het laatste teken uit de file gelezen is wordt
; de C-vlag 0
```

```
;
131B A6 A2      RDFIL   LDX      PCFILE      Laad het inputfilennummer
131D 20 00D0    JSR      SREAD      Single read uit file
1320 B0 19      BCS      RDCEND      Test op end of file
1322 C9 20      CMP      #$20      Test op een spatie
1324 F0 F5      BEQ      RDFIL      sla deze over
1326 C9 3B      CMP      #'      test op remark teken
1328 F0 04      BEQ      READREM
132A C9 23      CMP      ##      Test op scheidingsteken
132C 38      SEC
132D 60      RTS
132E A6 A2      READREM LDX      PCFILE      Laad het inputfilennummer
1330 20 00D0    JSR      SREAD      Single read uit file
1333 B0 06      BCS      RDCEND      Test op end of file
1335 C9 0D      CMP      #$0D      Test eol
1337 D0 F5      BNE      READREM      lees verder
1339 F0 E0      BEQ      RDFIL
133B A6 A2      RDCEND  LDX      PCFILE      Laad het Printerinputfilennummer
133D 20 48D0    JSR      CLOSE      Sluit de file
1340 18      CLC      C=0 voor einde file
1341 60      RTS
```

```
;
; File commandobuffer voor de printer-conversiefile
```

```
1342 533A504346 PCFBUFF FCC      'S:PCF.DAT',0
1347 2E44415400
```

```
;
; Tabel met speciale printer codes
```

```
134C      PCFTAB  RES      128      printercode tabel
;
1000      END      CONV
global labels
```

ASCERR	126F	ASCHEX	C03E	BUFINDX	00A4	CERROR	12B3	CLOSE	D048
CONV	1000	CONVPRINT	11F5	CREATE	D036	DATASAV	00A3	ENDPCF	124E
ERMES	DOB7	ERROR1	12B0	FILEIN	00A0	FILEOUT	00A1	INFO	100A
LOOPERR	1309	LSPCF	121F	NOSUBL	1267	OPEN	D03F	OUT	C023
PCFBUFF	1342	PCFCLO	12A0	PCFILE	00A2	PCFTAB	134C	PRINT	C03B
RDCEND	133B	RDEND	12A5	RDFIL	131B	READ	1253	READREM	132E
SREAD	D000	SUBLET	0012	SWRITE	D009				

Errors detected: 0

```

/*****
/*
/*      --- file GDP.C ---
/*
/*      simple driver for Thomson Grafic Display Processor EF9367
/*
/*      written by : Adri Hankel
/*      directed by : Erwin Visschedijk
/*
/*      copyright (c) 1988 : KIM Gebruikers Club Nederland.
/*
/*      attention: compiling this file (using 'ccc gdp')
/*                  takes more than eight minutes
/*                  (on a 1 MHz floppy system).
*****/

/* self-explaining */
#define TRUE 1
#define FALSE 0

/* maximum value for x & y coordinates */
#define MAX_X 511
#define MAX_Y 255

/* command key definitions */
#define UPKEY 11      /* key VER. TAB */
#define DOWNKEY 10    /* key LINEFEED */
#define LEFTKEY 8      /* key BACKSPACE */
#define RIGHTKEY 9     /* key HOR. TAB */
#define PLOTLINE 76    /* key 'L' */
#define CIRCLE_KEY 67  /* key 'C' */
#define BOX_KEY 66     /* key 'B' */
#define SELECT_PAGE 80 /* key 'P' */
#define ERASE_PAGE 69  /* key 'E' */
#define STEPSIZE 83     /* key 'H' */
#define CENTERCURSOR 72 /* key 'T' */
#define SETLINETYPE 84  /* key 'O' */
#define QUIT 81         /* key RETURN */
#define CR 13

/* assembler definitions */
#define asm
status equ $e150
cmd equ $e150
ctrl1 equ $e151
ctrl2 equ $e152
csize equ $e153
deltax equ $e155
deltay equ $e157
xmsb equ $e158
xlsb equ $e159
ymsb equ $e15a
ylsb equ $e15b
color equ $e164
scroll equ $e165
page equ $e166
hscroll equ $e167
#define endasm

/* global variables */
int curx,cury; /* current cursor coordinates */
int oldx,oldy; /* old cursor coordinates */
char plotflag; /* boolean: TRUE during plotting */
char curpage; /* current page */
char stepsize; /* number of steps (1..9) */
char linetype; /* linetype (continuous, dotted etc.) */
char old_linetype; /* temp. copy of linetype */

main()
{
char com_key;
if (yesgdp() == FALSE) /* check for gdp-pcb in the system */
    printf("*** Error : GDP not found, program terminated.\n");
    exit(1);
}

```

```

init_gdp();
init_screen();
dr_window();
do
{
    com_key = getkey();
    switch (com_key)
    {
        case UPKEY:
            reset_cursor();
            cury = cury + stepsize;
            set_cursor();
            show_status();
            break;
        case DOWNKEY:
            reset_cursor();
            cury = cury - stepsize;
            set_cursor();
            show_status();
            break;
        case LEFTKEY:
            reset_cursor();
            curx = curx - stepsize;
            set_cursor();
            show_status();
            break;
        case RIGHTKEY:
            reset_cursor();
            curx = curx + stepsize;
            set_cursor();
            show_status();
            break;
        case PLOTLINE:
            plotline();
            break;
        case CIRCLE_KEY:
            set_circle();
            break;
        case BOX_KEY:
            plot_box();
            break;
        case SELECT_PAGE:
            selec_page();
            break;
        case ERASE_PAGE:
            era_page();
            break;
        case STEPSIZE:
            set_stepsize();
            break;
        case CENTERCURSOR:
            center_cursor();
            break;
        case SETLINETYPE:
            se_linetype();
            break;
    }
}
while (com_key != QUIT);
clear_screen();
printf("End GDP\n");

plotline()
{
    char com_key;
    oldx = curx;
    oldy = cury;
    plotflag = TRUE;
    show_status();
    gotoxy(24,8);
    printf("Plotting.. (CR to end)");
    gotoxy(1,21);
    reset_cursor();
    do
    {
        com_key = getkey();

```

```

switch (com_key)
{
case UPKEY:
    unplot(oldx,oldy,curx,cury);
    cury = cury + stepsize;
    break;
case DOWNKEY:
    unplot(oldx,oldy,curx,cury);
    cury = cury - stepsize;
    break;
case LEFTKEY:
    unplot(oldx,oldy,curx,cury);
    curx = curx - stepsize;
    break;
case RIGHTKEY:
    unplot(oldx,oldy,curx,cury);
    curx = curx + stepsize;
    break;
}
plot(oldx,oldy,curx,cury);
show_status();
}
while (com_key != CR);
set_cursor();
gotoxy(24,8);
clreol();
plotflag = FALSE;
show_status();
}

set_circle()
{
int radius;
int max_rad;
char com_key;
max_rad = MAX_Y/2;
gotoxy(24,9);
clreol();
gotoxy(24,9);
printf("Enter radius (max. %d) : ",max_rad);
scanf("%d",&radius);
if (radius > max_rad) set_circle();
old_linetype = linetype;
linetype = 0;
set_linetype(linetype);
show_linetype();
circle(curx,cury,radius);
gotoxy(24,9);
clreol();
printf("Full circle (Y/N)");
com_key = getkey();
if (com_key == 89) /* com_key was 'Y' */
    full_circle(curx,cury,radius-1);
gotoxy(24,9);
clreol();
linetype = old_linetype;
set_linetype(linetype);
show_linetype();
}

plot_box()
{
char com_key;
oldx = curx;
oldy = cury;
plotflag = TRUE;
show_status();
gotoxy(24,10);
printf("Plotting box.. (CR to end)");
gotoxy(1,21);
reset_cursor();
do
{
com_key = getkey();
switch (com_key)

```

```

        case UPKEY:
            unbox(olddx,oldy,curx,cury);
            cury = cury + stepsize;
            break;
        case DOWNKEY:
            unbox(olddx,oldy,curx,cury);
            cury = cury - stepsize;
            break;
        case LEFTKEY:
            unbox(olddx,oldy,curx,cury);
            curx = curx - stepsize;
            break;
        case RIGHTKEY:
            unbox(olddx,oldy,curx,cury);
            curx = curx + stepsize;
            break;
    }
    box(olddx,oldy,curx,cury);
    show_status();
}
while (com_key != CR);
set_cursor();
gotoxy(24,10);
clreol();
gotoxy(24,10);
printf("Full box (Y/N)");
com_key = getkey();
if (com_key == 89) /* com_key was 'Y' */
{
    full_box(olddx,oldy,curx,cury);
}
gotoxy(24,10);
clreol();
plotflag = FALSE;
show_status();
}

selec_page() /* select a GDP page */
{
    char com_key;
    gotoxy(24,11);
    printf("Enter new page (0/1/2/3) :");
    com_key = getkey();
    if (com_key > 51) selec_page();
    if (com_key < 48) selec_page();
    curpage = com_key - 48;
    set_page(curpage);
    dr_window();
    set_cursor();
    gotoxy(24,11);
    clreol();
    show_status();
}

era_page() /* erase current GDP page */
{
    char com_key;
    gotoxy(24,12);
    printf("Erase this page ? (Y/N) :");
    com_key = getkey();
    if (com_key == 89) /* com_key was 'Y' */
    {
        clear_page(curpage);
        dr_window();
        set_cursor();
    }
    gotoxy(24,12);
    clreol();
    show_status();
}

set_stepsize()
{
    char com_key;
    gotoxy(24,13);
    printf("Enter stepsize (1-9) :");

```

```

com_key = getkey();
if (com_key > 57) set_stepsize();
if (com_key < 49) set_stepsize();
stepsize = com_key-48;
gotoxy(24,13);
clicol();
show_stepsize();

center_cursor()
{
char com_key;
gotoxy(24,14);
printf("Are you shure (Y/N)");
com_key = getkey();
if (com_key == 89) /* com_key was 'Y' */
    reset_cursor();
    curx = MAX_X/2; /* locate cursor at screen center */
    cury = MAX_Y/2;
    set_cursor();
gotoxy(24,14);
clicol();
show_status();
}

se_linetype() /* select a line type */
{
char com_key;
gotoxy(24,15);
printf("0=continuous, 1=dotted, 2=dashed, 3=dotted-dashed : ");
com_key = getkey();
if (com_key > 51) se_linetype();
if (com_key < 48) se_linetype();
linetype = com_key-48;
set_linetype(linetype);
gotoxy(24,15);
clicol();
show_linetype();
}

/*****
/* low level functions */

/* circle routine according to Bresenham/Mich algorithm */
circle(xstart,ystart,radius)
int xstart,ystart,radius;
{
int x,y,d;
x = 0;
y = radius;
d = 3-2*radius;
while (x < y)
{
    circle_points(xstart,ystart,x,y);
    if (d < 0)
    {
        d = d+4*x+6;
    }
    else
    {
        d = d+4*(x-y)+10;
        y--;
    }
    x++;
}
if (x == y)
{
    circle_points(xstart,ystart,x,y);
}
}

circle_points(xstart,ystart,x,y)
int xstart,ystart,x,y;

```

```

int corrx,corry;
corrx = (x*3)/2; /* adjust x & y to obtain a real circle, */
corry = (y*3)/2; /* instead of a oval */
wr_pixel(xstart+corrx,ystart+y);
wr_pixel(xstart+corrx,ystart-x);
wr_pixel(xstart+corrx,ystart-y);
wr_pixel(xstart-corr,x,ystart-y);
wr_pixel(xstart-corr,ystart-x);
wr_pixel(xstart-corr,ystart+y);
|

wr_pixel(xpos,ypos) /* write one pixel */
int xpos,ypos;
|
int xorga,yorga;
xorga = xpos;
yorga = ypos;
ready();
#asm
    lda    #0                ;deltay & deltax = 0
    sta    deltax            ;means setting one pixel only
    sta    deltax
    ldd.u  0                ;set X,Y memory addresses
    stx    ymsb
    sta    ylsb
    ldd.u  2
    stx    xmsb
    sta    xlsb
    lda    #$10              ;select pen
    sta    cmd
#endasm
|

full_circle(xstart,ystart,radius)
int xstart,ystart,radius;
|
do
    |
    circle(xstart,ystart,radius);
    radius--;
    |
while (radius != 0);
|

/* draw box between (xstart,ystart) and (xend,yend) */
box(xstart,ystart,xend,yend)
int xstart,ystart,xend,yend;
|
plot(xstart, ystart, xend, ystart); /* lower line */
plot(xstart, yend, xend, yend); /* upper line */
plot(xstart, ystart+1, xstart, yend-1); /* left line */
plot(xend, ystart+1, xend, yend-1); /* right line */
|

/* delete box between (xstart,ystart) and (xend,yend) */
unbox(xstart,ystart,xend,yend)
int xstart,ystart,xend,yend;
|
unplot(xstart, ystart, xend, ystart); /* lower line */
unplot(xstart, yend, xend, yend); /* upper line */
unplot(xstart, ystart+1, xstart, yend-1); /* left line */
unplot(xend, ystart+1, xend, yend-1); /* right line */
|

full_box(xstart,ystart,xend,yend)
int xstart,ystart,xend,yend;
|
int i;
if (yend > ystart)
    for (i = yend; i != ystart; i--)
        plot(xstart,i,xend,i);
|

```

```

else
    {
        for (i = ystart; i != yend; i--)
            plot(xstart,i,xend,i);
    }

plot (xorg,yorg,xend,yend) /* plot line from xorg,yorg to xend,yend */
{
    int xorg,yorg,xend,yend;
    select_pen();
    pen_down();
    setlin(xorg,yorg,xend,yend);
}

unplot (xorg,yorg,xend,yend) /* unplot line from xorg,yorg to xend,yend */
{
    int xorg,yorg,xend,yend;
    select_eraser();
    eraser_down();
    setlin(xorg,yorg,xend,yend);
}

setlin(xorg,yorg,xend,yend) /* set a single line */
{
    int xorg,yorg,xend,yend;
    int maxsize;
    int xtemp;
    int ytemp;
    maxsize = xend-xorg;
    abs(&maxsize);
    if (maxsize < 256)
        pline (xorg,yorg,xend,yend); /* single line plot possible */
}

else
    {
        if (yend == yorg) /* double line plot */
            ytemp = yorg;
        else
            ytemp = yorg + ((yend-yorg)/2);
        if (xend < xorg)
        {
            xtemp = xend + ((xorg-xend)/2);
            pline (xorg, yorg, xtemp, ytemp);
            xtemp--;
            pline (xtemp, ytemp, xend, yend);
        }
        else
        {
            xtemp = (xend-xorg)/2;
            pline (xorg, yorg, xtemp, ytemp);
            xtemp++;
            pline (xtemp, ytemp, xend, yend);
        }
    }

pline(xorg,yorg,xend,yend) /* plot line with delta_x < 256 */
{
    int xorg,yorg,xend,yend;
    int xorga,yorga;
    int dest; /* destination control */
    int deltax,deltay;
    deltax = (xend - xorg);
    abs(&deltax);
    deltax = (yend - yorg);
    abs(&deltay);
    xorga = xorg;
    yorga = yorg;
}

```

```

dest = 0x11;          /* starting value */
if (xend > xorg) dest = dest & 0xfd; /* compute sign bits */
                else dest = dest | 0x02; /* for DELTAX */
if (yend > yorg) dest = dest & 0xfb; /* and DELTAY */
                else dest = dest | 0x04;

ready();
#asm
    ldd.u 0
    sta    deltay
    ldd.u 2
    sta    deltax
    ldd.u 6
    stx    ymsb
    sta    ylsb
    ldd.u 8
    stx    xmsb
    sta    xlsb
    ldd.u 4
    sta    cmd
#endasm

init_gdp() /* initialize gdp processor */
{
    char i;
    for (i = 4; i > 0; i--)
        clear_page(i);
    curpage = 0;
    stepsize = 5;
    set_page(curpage);
    linetype = 0;
    old_linetype = 0;
    curx = MAX_X/2; /* locate cursor at screen center */
    cury = MAX_Y/2;
    oldx = curx;
    oldy = cury;
    plotflag = FALSE;
    select_eraser();
    pen_down();
    set_wr_normal();
    set_non_cyclic();
    set_continuous_line();
    set_normal_characters();
    set_horizontal_axis();
    set_cursor();
#asm
    ldx    #$00
    stx    color
    stx    scroll
    stx    hscroll
    stx    xmsb
    stx    xlsb
    stx    ymsb
    ldx    #$11
    stx    csize
    ldx    #$f8
    stx    ylsb
#endasm

abs(a) /* make a positive */
int *a;
{ if (*a < 0) *a = -*a; }

set_cursor()
{
    old_linetype = linetype;
    set_linetype(0);
    select_pen();
    pen_down();
    wr_pixel(curx, cury);
    linetype = old_linetype;
    set_linetype(linetype);
}

```

```

reset_cursor()
{
    old_linetype = linetype;
    set_linetype(0);
    select_eraser();
    pen_down();
    wr_pixel(curx,cury);
    linetype = old_linetype;
    set_linetype(linetype);
}

set_linetype(type)
char type;
switch (type)
{
    case 0:
        set_continuous_line();
        break;
    case 1:
        set_dotted_line();
        break;
    case 2:
        set_dashed_line();
        break;
    case 3:
        set_dot_dashed_line();
        break;
}

}

dr_window()
{
    old_linetype = linetype;
    set_linetype(0);
    boxT0,0,MAX_X,MAX_Y);
    linetype = old_linetype;
    set_linetype(linetype);
}

/*****
/* primitive functions */
/* CTRL1 register */

pen_up()
{
    #asm
        lda    ctrl1
        and    #$fe
        sta    ctrl1
    #endasm
}

eraser_up()
{
    #asm
        lda    ctrl1
        and    #$fe
        sta    ctrl1
    #endasm
}

pen_down()
{
    #asm
        lda    ctrl1
        ora    #$01
        sta    ctrl1
    #endasm
}

eraser_down()
{
    #asm
        lda    ctrl1
        ora    #$01
        sta    ctrl1
    #endasm
}

```

```

#endasm
|
select_pen()
#asm
    lda    ctrl1
    ora    #$02
    sta    ctrl1
#endasm
|
select_eraser()
#asm
    lda    ctrl1
    and    #$fd
    sta    ctrl1
#endasm
|
set_wr_normal()
#asm
    lda    ctrl1
    and    #$fb
    sta    ctrl1
#endasm
|
set_wr_highspeed()
#asm
    lda    ctrl1
    ora    #$04
    sta    ctrl1
#endasm
|
set_non_cyclic()
#asm
    lda    ctrl1
    and    #$f7
    sta    ctrl1
#endasm
|
set_cyclic()
#asm
    lda    ctrl1
    ora    #$08
    sta    ctrl1
#endasm
|
/* CTRL2 register */
set_continuous_line()
#asm
    lda    ctrl2
    and    #$fc
    sta    ctrl2
#endasm
|
set_dotted_line()
#asm
    lda    ctrl2
    ora    #$01
    sta    ctrl2
#endasm
|

```

```

set_dashed_line()
#asm
    lda    ctrl2
    ora    #$02
    sta    ctrl2
#endasm
|
set_dot_dashed_line()
#asm
    lda    ctrl2
    ora    #$03
    sta    ctrl2
#endasm
|
set_normal_characters()
#asm
    lda    ctrl2
    and    #$fb
    sta    ctrl2
#endasm
|
set_tilted_characters()
#asm
    lda    ctrl2
    ora    #$04
    sta    ctrl2
#endasm
|
set_horizontal_axis()
#asm
    lda    ctrl2
    and    #$f7
    sta    ctrl2
#endasm
|
set_vertical_axis()
#asm
    lda    ctrl2
    ora    #$08
    sta    ctrl2
#endasm
|
ready()
#asm
    lda    status
    and    #$04
    beq    _ready
#endasm
|
clear_page(page)
char page;
#asm
    ldc.u    0    page
    sta    page
    lda    #$07
    sta    cmd
#endasm
|
    ready();
|
set_page(page)
char page;

```

```

{asm
    ldc.u    0
    sta      page
#endasm

getkey() /* get single key from keyboard without echo to screen */
#asm
    jsr      $f003
    jsr      $c041      toupper
    std.u    0
#endasm

yesgdp() /* checks on available GDP pcb */
#asm
    lda      cmd
    and      #$f0
    bne      nogdp      ;no gdp
    ldd      #1          ;return(1)
    rts
nogdp      ldd      #0          ;return(0)
    rts
#endasm

/***** screen procedures *****/
show_status() /* shows actual settings */
{
    show_xy();
    gotoxy(1,22);
    printf("Page=%d",curpage);
    gotoxy(1,21);
}

show_xy() /* shows current x & y parameters */
if (plotflag == TRUE)
{
    gotoxy(41,22);
    printf("Oldx=%d ",oldx);
    gotoxy(51,22);
    printf("Oldy=%d ",oldy);
}
else
{
    gotoxy(41,22);
    clrcol();
    gotoxy(61,22);
    printf("Curx=%d ",curx);
    gotoxy(71,22);
    printf("Cury=%d ",cury);
}

show_stepsize()
{
    gotoxy(10,22);
    printf("Stepsize=%d",stepsize);
    gotoxy(1,21);
}

show_linetype()
{
    gotoxy(23,22);
    clrcol();
    printf("Lntype=");
    switch (linetype)
    {
        case 0: printf("Contin.");
    }
}

```

```

        case 1: break;
               printf("Dotted");
               break;
        case 2: printf("Dashed");
               break;
        case 3: printf("Do./da.");
               break;
    }
    show_status();
}

clear_screen()
{
    printf("%c",12);
}

inv_on()          /* set inverse */
{
    printf("%c%c",27,'i');
}

inv_off()         /* reset inverse */
{
    printf("%c%c",27,'n');
}

gotoxy(x,y)       /* cursor to x,y. Adapted from A. Megens */
char x,y;         /* video routines */
#asm
    ldc.u    0
    pha
    ldc.u    2
    tay
    pla
    tax
    jsr      $f024
#endasm

clreol()          /* clear until end of line */
{
    printf("\032");
}

init_screen()
{
    clear_screen();
    inv_on();
    printf("Simple GDP driver ");
    printf("v1.0");
    inv_off();
    printf("\n ^ - cursor up      |\n");
    printf("\n v - cursor down      |\n");
    printf("\n < - cursor left       |\n");
    printf("\n > - cursor right      |\n");
    printf("\n L - plot Line         |\n");
    printf("\n C - plot Circle       |\n");
    printf("\n B - plot Box          |\n");
    printf("\n P - select Page       |\n");
    printf("\n E - Erase page        |\n");
    printf("\n S - set Stepsize      |\n");
    printf("\n H - cursor Home       |\n");
    printf("\n T - select lineType   |\n");
    printf("\n O - Quit GDP          |\n");
    gotoxy(1,20);
    printf("-----");
    printf("-----\n");
    show_linetype();
    show_status();
    show_stepsize();
}

/* end gdp.c */

```

### COMPUTER GRAPHICS

#### DISPLAY TECHNIEK:

De ontwikkeling van de display techniek heeft niet altijd gelijke tred gehouden met de ontwikkeling van de computer. Echter vooral de laatste jaren is daar grote verandering ingekomen ten gunste van de grafische display techniek. Het displayen van grafische informatie wordt in vele toepassingen gebruikt, zowel doormiddel van zeer kostbare industriële systemen, als door de steeds goedkoper wordende PC.

In het midden van de jaren zestig kwam de opmars van het beeldscherm toegepast als man-machine interface. De nieuwe term 'TELETYPE REPLACEMENT' was geboren. Luchtvaart reserverings-systemen waren de eerste grote toepassingen van de zeer kostbare VDU, echter net als met de calculators zakten de prijzen dramatisch en nam het gebruik sterk toe. De term 'VDU' (Video Display Unit) kwam in gebruik, vooral laat in de jaren 70 met de intrede van de z.g. intelligente terminal die er toe bijdroeg, dat het gebruik van computers gemeengoed is geworden. Begin jaren 80 werd de VDU een integraal onderdeel van de meeste microcomputer en personal computers.

Wat gebeurde er met de ontwikkeling van de z.g. computer graphics displays, in de tijd dat de alphanumeric beeldschermen zo succesvol in gebruik toenamen?

Computer graphics displays bestaan zelfs langer dan de eerder beschreven toepassingen. Tijdens de zestiger jaren waren ze vooral in gebruik in de wetenschap, techniek en militaire systemen. De toenmalige techniek stond bekend als stroke-writing (streepjes trekken) alternatieve namen waren 'randomscan, vector-graphics en calligraphics'. De 'stroke writing' schermen werkten met een adressen matrix te weten een startadres X,Y en een eind adres XY waar tussen alleen horizontale strepen getrokken konden worden. In de jaren zeventig zien we dat grafische techniek ook op andere gebieden toegepast word zoals procescontrole. De toen gebruikte techniek leek op die van VDU beeldschermen. Men maakte nl gebruik van 'semi graphics' ook wel block of format graphics genoemd, dit zijn symbolen die aaneengesloten (dus zonder spaties) geplaatst kunnen worden zodat een compleet plaatje ontstaat. (Zie figuur 1)

Normaal caraceter matrix.



Semie grafische caraceter matrix



Fig. 1

#### KLEUR:

Fabrikanten brachten begin jaren zeventig de eerste industriële kleuren monitoren op de markt, de resolutie was nog laag en ergonomisch mensvriendelijk waren deze monitoren bepaald nog niet. De toepassing van kleuren monitoren gekombineerd met semie-grafische weergave nam desalniettemin grote vormen aan, en leidde tot grote verbeteringen bij deze monitoren. De basis kleuren bij monitoren zijn: ROOD GROEN en BLAUW "RGB" Door deze kleuren te combineren en de intensiteit per kleur te regelen, kunnen we een zeer groot kleuren palet op de monitor weergeven, bijvoorbeeld:

INTENSITEIT	ROOD	GROEN	BLAUW	KLEUR
1	0	0	0	ZWART
1	0	0	1	BLAUW
1	0	1	0	GROEN
1	0	1	1	CYAAN
1	1	0	0	ROOD
1	1	0	1	IVIOLET
1	1	1	0	GEEL
1	1	1	1	WIT
0 HALF INT.	0	0	0	ZWART
0 HALF INT.	0	0	1	BLAUW
0 HALF INT.	0	1	0	GROEN
0 HALF INT.	0	1	1	CYAAN
0 HALF INT.	1	0	0	ROOD
0 HALF INT.	1	0	1	IVIOLET
0 HALF INT.	1	1	0	BRUIN
0 HALF INT.	1	1	1	GRIJS

1 = AKTIEF

Fig. 2

Zoals u zult hebben gemerkt, kunnen we door de intensiteit om te zetten naar halfintensiteit 16 kleur combinaties genereren, bij monitoren met een z.g. TTL ingang voor de RGB aansluitingen, is dit meestal het maximaal aantal Kleuren dat men kan weergeven. Bij monitoren met een analoge ingang voor de RGB aansluitingen zijn het aantal weer te geven kleur-combinaties bijna oneindig en meer afhankelijk van de kontroler die men gebruikt. Vergelijkt u het maar met een lichtschakelaar, aan/uit voor TTL en een lichtschakelaar met dimregelaar voor analoog.

### FULL GRAPHICS:

Deze techniek van weergeven, is een van de snelst groeiende o.a door de dalende prijzen van hoge resolutie-monitoren en de bijbehorende controllers. Wat is FULL GRAPHICS? Het is een techniek die het mogelijk maakt, op een willekeurige positie op het beeldscherm een "PUNT" weer te geven. Deze punt noemt men in de regel "PIXEL" of ook wel "DOT". Het maximale aantal pixels die men horizontaal (X) en vertikaal (Y) kan weergeven, is afhankelijk van ondermeer het beeldscherm. Men noemt dat de "RESOLUTIE". Hoe hoger de resolutie is, des te sneller zal de monitor moeten zijn. De snelheid van een monitor wordt uitgedrukt in de video bandbreedte welke kan oplopen tot meer dan 100 MHZ. Uiteraard is ook de horizontale synchronisatie van grote betekenis, de snelheid daarvan wordt meestal uitgedrukt in een min en max waarde, uitgedrukt in KHZ. Het is u wel opgevallen dat een beeldscherm niet symmetrisch is, maar in de X richting (horizontaal) langer is dan in de Y richting (vertikaal), daarom is het gebruikelijk om de resolutie asymmetrisch en in verhouding tot het beeldscherm op te geven. Gebruikelijke resoluties zijn:

1280 X 1024 pixels

1024 X 768 pixels

768 X 512 pixels

720 X 400 pixels

720 X 350 pixels

640 X 480 pixels

640 X 400 pixels

640 X 350 pixels

640 X 200 pixels

320 X 200 pixels

Het systeem of controller die de pixels opslaat, en op kommando naar het beeldscherm kan sturen moet een geheugen hebben dat minimaal overeenkomt met de resolutie die het beeldscherm weergeeft. Zo'n geheugen noemt men wel "BITMAP" of "VIDEO GEHEUGEN" en bestaat uit snelle dynamische-ram chips. Om een beeldscherm volledig te vullen met pixels in een resolutie van laten we zeggen 1280 X 1024 pixels, is een minimaal geheugen nodig van 1310720 bits of 163840 bytes (164 K) Met dit geheugen kunnen we op iedere plaats op het beeldscherm een pixel aan of uit zetten. Jammer is, dat het meer in een kleur kan. Voor meer kleuren hebben we ook meer geheugen nodig, maar met de tegenwoordige techniek op het gebied van geheugen chips, is dat probleem goed opgelost. (denk aan de mega-chip)

### MEER KLEUR:

Voor een bepaalde resolutie, kunnen we de grote van de bitmap uitrekenen, maar slechts voor 1 kleur. Een derde dimensie gaan we nu toevoegen, namelijk meer bitmaps. Het aantal bitmaps tot de macht 2 bepaald het aantal kleuren dat we kunnen weergeven, dus 1280 X 1024 X 2 betekent twee bitmaps van 1280 X 1024 nu kunnen we 2 tot de macht 2=4 kleuren weergeven. Zie figuur 3.

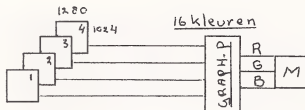


Fig. 3

Vier bitmaps is een zeer populair aantal men kan er 16 kleuren mee displayen, dat zal voor de meeste toepassingen voldoende zijn. Echter wil men b.v. de mogelijkheden van een analoge monitor volledig benutten dan moet men zeker 256 kleuren kunnen aansturen. U kunt nu zelf uitrekenen hoeveel bitmaps er nodig zijn nl. 1280 X 1024 X 8. Het totale geheugen bedraagt 1.32 Mb. Misschien bent u wel eens in een specificatie tegengekomen de uitdrukking 256 kleuren uit een PALET van 262144 kleuren, of 64 grijs tinten. Het principe hiervan is dat men een kleuren-tabel (load color table) die zo groot is als de binaire combinaties van het kleuren register, dit register noemt men nu "KLEUREN TABLET". In ons voorbeeld zijn er 262144 kleuren geladen, hieruit kunnen 256 kleuren worden gekozen, en vervolgens worden weergegeven op de monitor, precies zoals de kunstschilder zijn kleuren kiest van zijn schilders-palet. Het kleuren register ziet er als volgt uit:

RRRRRR	GGGGGG	BBBBBB	18 bits
--------	--------	--------	---------

Het zijn 6 bits rood 6 bits groen 6 bits blauw, en zoals eerder omschreven kunnen wij met 1 bit van de kleuren RGB zonder intensiteit-bit 8 kleuren maken, dus kunt u zelf uitrekenen dat 18 bits RGB 262144 kleuren combinaties opleveren. Voor grijs-tinten moeten we altijd tenminste 1 bit rood 1 bit groen en 1 bit blauw gebruiken dus zijn er slechts 64 combinaties mogelijk. Over Grafische software-tools zoals GKS (Graphic Kernel System) heb ik nog niet gesproken, misschien dat ik over dat onderwerp nog eens een apart artikel publiceer.

Rinus Vleesch Dubois.

### DE VIDEO CONTROLLER CRT 9028/9128.

P. de Vries  
Karperstraat 12  
5022 HJ Tilburg  
013-355311  
(Door de redactie bewerkt).

De CRT 9028 is geschikt voor computersystemen met als CPU een 8085, 8086, Z80 etc. De CRT 9128 is geschikt voor de CPU-types 6502, 68000, Z8 etc. Verder zijn de chips nog onderverdeeld in diverse functietypes met diverse beeldformaten, aantal dots per teken enzovoort. Het IC is masker-programmeerbaar waardoor de diverse functietypes door de fabrikant op eenvoudige wijze ingeprogrammeerd kunnen worden.

De CRT 9128-0000 produceert 80 tekens per regel, 24 regels + een statusregel per beeld.

### Mogelijkheden.

De chip heeft de volgende mogelijkheden:

#### Video-attributen:

Teken-onafhankelijke attributen zoals cursor-blok of -lijntje, cursor-onderdrukking en inverse video.  
Teken-afhankelijke attributen zoals teken-onderdrukking, teken reverse, teken-onderlijning en helderheidsbeïnvloeding van een teken.

#### Alfanumerieke en grafische mode:

In de alfanumerieke mode kunnen de video-attributen werkzaam zijn. In de grafische mode kunnen we drie types onderscheiden. In de eerste plaats is dat een alfanumerieke mode die zich van de normale alfanumerieke mode onderscheidt in het feit dat dat de video-attributen niet werkzaam zijn, ook al zijn ze door het programma geactiveerd. Verder zijn er een smalle en een brede grafische mode. De grafische tekens in deze modes zijn opgedeeld in segmenten die in of uitgeschakeld kunnen worden. In de onderstaande figuren is aangegeven hoe deze segmenten over het teken verdeeld zijn:

	7	6	5	4	3	2	1		7	6	5	4	3	2	1
0	.	.	.	.	.	.	.	0	6	6	6	6	3	3	3
1	.	.	.	.	.	.	.	1	6	6	6	6	3	3	3
2	.	.	.	.	.	.	.	2	6	6	6	6	3	3	3
3	.	.	.	.	.	.	.	3	5	5	5	5	2	2	2
4	.	.	.	.	.	.	.	4	5	5	5	5	2	2	2
5	4	4	4	X	2	2	2	5	5	5	5	2	2	2	2
6	.	.	.	.	.	.	.	6	5	5	5	5	2	2	2
7	.	.	.	.	.	.	.	7	4	4	4	4	1	1	1
8	.	.	.	.	.	.	.	8	4	4	4	4	1	1	1
9	.	.	.	.	.	.	.	9	4	4	4	4	1	1	1

### Scrolling:

Het IC heeft zowel een jump scroll als een smooth scroll mogelijkheid. De jump scroll is de normale scrollprocedure, dus regels worden in hun geheel naar boven of naar beneden over het scherm bewogen. Bij smooth scroll worden de regels steeds met enkele video-lijnen tegelijk verplaatst, totdat het hele beeld een regel opgeschoven is.

### Statusregel:

Voor functietypes die 25 regels per beeld hebben, is de statusregel aan- en uitschakelbaar.

### Schakeling in een 6502-systeem.

Uit fig. 7 blijkt, dat de schakeling zeer eenvoudig is en maar weinig componenten nodig heeft. Bekijkt men het blokschema van de chip (fig. 1), dan ziet men direct dat de character-generator, het video-schuifregister en de datamultiplexer al in het IC zijn ingebakken. Gaan we terug naar fig. 7, dan zien we dat de CPU, via D2 (CRT 9128) data van of naar het video-RAM D3 transporteert. D1 is een adresdecoder die gemaakt kan worden met NANDs, comparatoren enz. enz., de keuze hangt af van hoe het systeem is opgebouwd. Hetzelfde geldt voor D3. Hiervoor kan elke inverter-buffer gebruikt worden mits deze voldoende stroom kan leveren om het weerstandennetwerk te kunnen aansturen. Ik heb de volgende IC's gebruikt: 3 \* 74HC85 (4 bits magnitude comparator), 1/2 \* 74HC20 (dual four input nand gate) en voor D3 1 \* 74HC240 (octal inverter buffer).

### Programmeren van het IC.

Het programmeren van het IC bespreek ik met behulp van de figuren 2 t/m 6c.

#### Figuur 2: Registeradresering.

Hier is niet veel meer over te vertellen dan dat het op dezelfde wijze gebeurt als bij de 6845 of de 6545. Zoals men in figuur 2 ziet, is het dataregister geschikt om in te schrijven en uit te lezen. Dit is echter niet geheel waar want bekijkt men figuur 4, dan ziet men dat alleen het character-register deze eigenschap heeft. In de overige registers kan men alleen data schrijven en niet eruit lezen. Verder kan men in het adresregister alleen schrijven en men kan alleen lezen uit het statusregister.

#### Figuur 3: Statusregister.

Bit 7 van het statusregister, het DONE-bit is alleen van belang, de rest is don't care. Is bit 7 een "1", dan is de video-

chip bezig met videobesturing. Is dit bit een "0", dan staat het IC dataverandering toe in de registers CURH1, CURLO en CHARACTER. In de overige registers mag men de data veranderen, ongeacht in welke toestand bit 7 zich bevindt.

### Figuur 4: Dataregisters.

In deze figuur is een opsomming gegeven van de acht dataregisters met daarbij of er in het register lees en/of schrijfoperaties toegestaan zijn. Men selecteert een gewenst dataregister als volgt:

```
LDA #06      ; adres Reset registers
STA ADRES    ; zet in adresregister
LDA #20      ; dummy teken (spatie)
STA DATA    ; chip is nu gereset
```

Hetzelfde geldt voor de overige dataregisters met uitzondering van de cursor- en characterregisters, dan laat men de bovenstaande instructies vooraf gaan door:

### BERGOP:

```
BIT STATUS    ; bit 7 statusreg. 0?
BMI BERGOP    ; nee, wacht dan
```

### Figuur 5 en 6: Dataregisters.

### CHIPRESET

Het IC wordt gereset als er een willekeurig teken in wordt geschreven.

### TOSADD

Het T1M-bit is een rasterscan-mode bit; T1M=1 50 Hz, T1M=0 60 Hz. Bit 6 t/m 0 bepalen het startadres in het videoram voor de top van het scherm. Bit 6 t/m 0 bepalen sturen hierbij de lijnen DA10 t/m DA0 aan. De lijnen DA3 t/m DA0 worden door de chip op nul gezet zodat alleen de startadressen 0, 16, 32 enz. tot 1920 mogelijk zijn. De statusregel 1920-1999 kan niet als beginadres geadresseerd worden.

### CURLO

Is een normaal cursor-register, die op de gebruikelijke wijze wordt geladen. Bit 7 t/m bit 0 bepalen het lage byte van het cursoradres DA7 t/m DA0.

### CURH1

Bevat de rest van het cursoradres namelijk bit 2 t/m 0 resp. DA10 t/m DA8. Bit 7, SLE, beïnvloedt de statusregel. SLE=1 statusregel is aan en doet niet mee als er een scroll optreedt. SLE=0 statusregel is uit maar nu scrollt de statusregel wel mee. Echter, men kan nu geen tekens op de

statusregel zetten. Bit 6 t/m bit 3, SS3 t/m SS0, hebben invloed op de scrolling mode. Zet men deze bits op 0, dan treedt er een jump-scroll op als het scherm geheel is volgeschreven met 24 regels. SS3 t/m SS0 bevatten in binaire code de eerste video-lijn die bovenaan het scherm staat. Verhoogt men deze code met X, dan zal het beeld X videolijnen naar boven bewegen. Op deze manier kan men een smooth-scroll uit laten voeren.

### FILADD

Is het vuladresregister. Met behulp van dit register kan men het scherm geheel of gedeeltelijk wissen. Bit 6 t/m bit 0 = DA10 t/m DA4 bevatten het eindadres, het cursoradres is het beginadres van het te wissen gedeelte van het scherm. Men hoeft slechts één maal een spatie (\$20) in het CHARACTER-register te zetten om het gewenste gedeelte in zijn geheel te wissen. Ook hier worden de lijnen DA3 t/m DA0 op nul gehouden en mag ook niet het adres van de statusregel ingevuld worden.

### ATTDAT

Het attribuutregister. Bit 7 = '0' betekent de alfanumerieke mode, bit 7 = '1' de grafische mode. Als bit 6 t/m bit 0 allen '0' zijn, dan zijn alle attributen uitgeschakeld. Bit 1 echter één of meer van deze bits '1', dan zijn de volgende attributen actief.

A: de teken-onafhankelijke attributen.

Bit 6: De cursor wordt onderdrukt. Door dit bit afwisselend '0' en '1' te maken, kan met de cursor laten knipperen.

Bit 5: De cursor is een lijntje, normaal is het een blokje.

Bit 4: Inverse video.

B: De teken-afhankelijke attributen. Om deze te laten werken, moet bovendien het TAG-bit van het CHARACTER-register op '1' gezet worden.

Bit 3: Tekens-onderdrukking. Hiermee kan men dus een teken laten knipperen. Ook kan men een geheime tekst op het scherm zetten die een ander bijvoorbeeld niet mag lezen.

Bit 2: Intensiteit d.w.z. het teken wordt helderder dan de andere tekens indien men INT-OUT (fig. 7) op de aangegeven wijze aanstuurt. Laat men

een inverter weg, dan wordt het teken minder helder dan de overige tekens.  
 Bit 1: Het teken wordt onderlijnd.  
 Bit 0: Het teken staat inverse t.o.v. de andere tekens.

## CHARACTER

Dit register bevat een TAG-bit, bit 7, en nog zes andere bits die een mode-afhankelijke betekenis hebben.

A: Alfanummerieke mode (bit 7 in het attribuut = 0).

Is het TAG-bit nu een '1' dan worden de gekozen teken-afhankelijke attributen in het attributiek-register actief. In bit 6 t/m 0 is het ASCII-teken gecodeerd. Ook de controletekens zijn toegestaan en worden als een teken op het scherm weergegeven.

B: Grafische mode (bit 7 in het attribuut-register = 1).

Als het TAG-bit = '1', dan zit men toch weer in een alfanumerieke mode, echter nu met dien verstande dat de attributen nu niet meer werkzaam zijn. Is het TAG-bit = '0', dan zit men pas echt in de grafische mode. Hierin geeft W/T (bit 6) aan of men in de brede of in de smalle grafische mode zit. De rest van de bits geeft dan aan welke segmenten aan en uitgeschakeld moeten worden.

Bij het experimenteren met het IC ben ik enkele eigenaardighedenheden tegengekomen. Het blijkt dat als bit 7 van CURHI '0' is, de statusregel inderdaad niet te zien is als het beginadres 0 is. Is echter het beginadres ongelijk aan nul, dan is de statusregel wel zichtbaar maar de tekst die, volgens mij, nu op regel 24 staat is niet te zien. Pas als de hele regel volgeschreven is of als men via een return een scroll uit laat voeren, dan is de onzichtbare regel zichtbaar geworden. Uit de datasheet heb ik niet op kunnen maken dat het IC zo zou moeten functioneren en hoe het IC anders geprogrammeerd zou moeten worden want men wil toch graag zien wat men intypt en niet pas na een scroll.

Een tweede eigenaardigheid ontdekte ik toen ik de cursor op de statusregel gezet had en per ongeluk het commando wis scherm vanaf de cursorpositie tot het einde ingegeven had in plaats van het commando om de statusregel te wissen. De video-processor wiste het hele scherm, met inbegrip van de statusregel en de cursor. Verder bleek dat hij op geen enkel commando meer reageerde, zelfs niet meer op reset. Nadat de voedingsspanning enkele tijd uitgeschakeld geweest was, reageerde de processor weer normaal.

De grafische mogelijkheden heb ik nog niet geprobeert omdat ik eerst mijn computersysteem verder op wil bouwen en omdat je voor normaal programmeren de grafische mode niet echt nodig hebt.

### Ervaringen.

Om met de chip te kunnen experimenteren, heb ik het IC opgenomen in mijn ALM-65 systeem. Verder werd er een apart toetsenbord waarop ook de kleine letters ingegeven kunnen worden op het systeem aangesloten. Door middel van een programma dat afgeleid is van de 10-65 uit DOS-65 kan het IC aangestuurd worden.

Persoonlijk vind ik de CRT 9128 een prachtig LC. Ik vind het alleen vervelend dat als ik de statusregel uitschakel ik niet meer kan zien wat er ingetypt wordt. Ik heb van alles geprobeerd maar kom er niet uit. Misschien dat ~~een~~ van de andere leden mij kan helpen.

Het LC is o.a. verkrijgbaar bij de firma Aurima in Eindhoven, telefoon 040-816565.

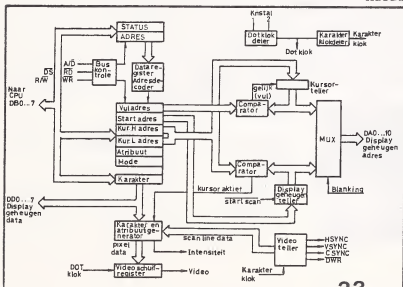


Fig.1

A/D	US	R/W	Register operatie
0	0	0	Schrijf in dataregister
0	0	1	Lees uit dataregister
1	0	0	Schrijf in adresregister
1	0	1	Lees uit statusregister

Fig. 2

B7	B6	B5	B4	B3	B2	B1	B0
DONE	X	X	X	X	X	X	X

FIG. 3

Data adresregister								Geader registreer	
B7	B6	B5	B4	B3	B2	B1	B0	TYPE	Registernaam
X	X	X	X	0	1	1	0	schrij	SCHIPSEET
X	X	X	X	1	0	0	0	schrijf	TDSADD
X	X	X	X	1	0	0	1	schrijf	CURLO
X	X	X	X	1	0	1	0	schrijf	CURHI
X	X	X	X	1	0	1	1	schrijf	FILADO
X	X	X	X	1	1	0	0	schrijf	ATTDAT
X	X	X	X	1	1	0	1	S/L	CHARACTER
X	X	X	X	1	1	1	0	schrijf	MODE REGISTER

FIG. 4

### DATA REGISTERS

B7	B6	B5	B4	B3	B2	B1	B0
CHIPRESET							
Dummy karakter							
CURLO kursoradresregister laag							
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
FILADD vuladresregister							
X	DA10	DA9	DA8	DA7	DA6	DA5	DA4
CHARACTER karakterregister							
TAG	Mode afhankelijk						

B7	B6	B5	B4	B3	B2	B1	B0
TQ5ADD Startadresregister							
TIM	DA10	DA9	DA8	DA7	DA6	DA5	DA4
CURHI kursoradresregister hoog							
SLE	SS3	SS2	SS1	SS0	DA10	DA9	DA8
ATTDAT atribuutdata register							
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
MODEREGISTER							
MOD	X	X	X	X	X	X	X

FIG.5

Grafische mode

CHARACTER register	
1	Alfanumeriek karakter

FIG.6a

CHARACTER register	
0	W/T SEG6 SEG5 SEG4 SEG3 SEG2 SEG1

FIG.6b

Alfanumerieke mode

CHARACTER register	
TAG	Alfanumeriek karakter

Karakter afhankelijke atribuut  
TAG=0; ult TAG=1; aan

FIG.6c

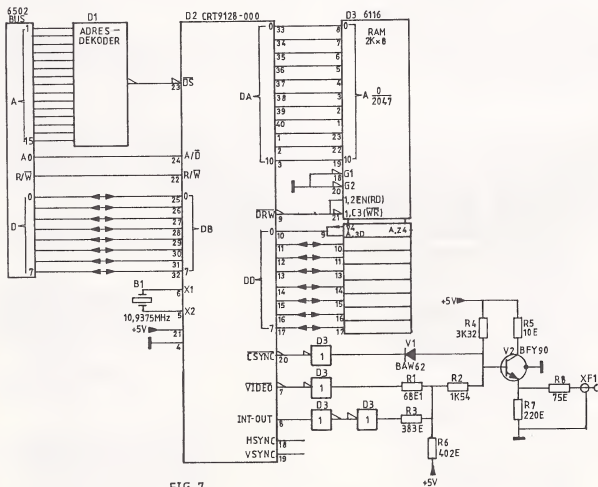


FIG.7

## DE VIDITEL KARAKTERGENERATOR (2)

B. de Bruine 5-3-88

Met de viditel karaktergenerator is het mogelijk om een grafisch scherm van 160x72 pixels op te bouwen. Hoe dit in zijn werk gaat beschrijft dit artikel.

### Indeling van het grafische scherm

Voor tekstgebruik is het scherm ingedeeld in 80 horizontale, en 24 verticale vakjes. (fig. 1) Op iedere positie kan men een karakter printen, bijvoorbeeld een letter of een cijfer. Door nu een blockgraphic te printen bestaande uit 2 horizontale en 3 verticale eenheden, word de resolutie vergroot tot 160 bij 72.

### Het plotten van een beeldpunt

Het leuke van de viditelkaraktergenerator is de logische indeling. Ieder beeldpunt in de bitmap (fig. 3) heeft een waarde (fig 5.)

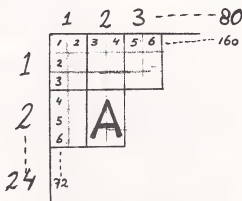



Fig. 1  
Schermindeling pixels

Zo heeft  de waarde 1, en  de waarde 2. Een spatie heeft

gewicht 0 en een inverse spatie 63. Wil men nu  printen, dan kan deze graphic samengesteld worden uit de pixels (1,1) en (2,1). Dat betekend:

$$\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}$$

$$\text{oftewel } 1 + 2 = 3$$

$$\text{dat is } 0000 \ 0001 \ \text{OR} \ 0000 \ 0010 = 0000 \ 0011$$

Grafische coördinaten (160,72)



Tekstcoördinaten (80,24)



Bitmapcoördinaten (2,3)

### Fig. 2 printmethode pixels

cursorbesturingscommando (b.v. CURSOR TO in basic) de cursor kan plaatsen en de berekende blockgraphic kan printen. De conversie van grafische- naar tekstcoördinaten binnen een scherm gaat als volgt:

$$HO = \text{INT}((GHO+1)/2)$$

$$VE = \text{INT}((GVE+2)/3)$$

Om de positie van het beeldpunt binnen de elementaire matrix te bepalen gebruiken we onderstaande formules:



Fig. 4 Demo

```
if GH0 > 2 then
    repeat MHO=GH0-2
    until MHO<=2
```

```
if GVE > 3 then
    repeat MVE=GVE-3
    until MVE<=3
```

GH0=grafisch horizontaal  
GVE=grafisch vert.

MHO=matrix hor.  
MVE=matrix vert.

HO=tekst horizontaal  
VE=tekst vertikaal

1,1	2,1
1,2	2,2
1,3	2,3

Fig. 3 Bitmap van de basismatrix

G1		0000 0001	1	QM(1,1)	\$01
G2		0000 0010	2	QM(2,1)	\$02
G3		0000 0100	4	QM(1,2)	\$04
G4		0000 1000	8	QM(2,2)	\$08
G5		0001 0000	16	QM(1,3)	\$10
G6		0010 0000	32	QM(2,3)	\$20

Fig. 5 De waarden van de zes mogelijke pixels in de basismatrix.

MHO,MVE geeft de positie aan in de basismatrix. Wat hierboven gebeurd is niets meer dan het terugbrengen van de beeldcoördinaten totdat de kleinstmogelijke matrixcoördinaten gevormd zijn. Door voor horizontaal er steeds twee, en vertikaal er steeds drie af te trekken ontstaat een matrix met basiscoördinaten zoals beschreven in fig. 5. Hiermee is ook de positie van het beeldpunt vastgelegd.

Voorbeeld: Plot een punt op GH0=7, GVE=5

→ H0 1 2 3 4 MHO= 7-2-2-2=1  
→ GH0 1 2 3 4 5 6 7 8 MVE= 5-3=2  
basiscoördinaten (1,2)

1							
2							
3							
4							
5							
6							

VE GVE  
↓ ↓

Fig. 6

Pixel op leeg scherm

Vervolgens berekenen we de tekstcoördinaten:  
HO=INT((GH0+1)/2) = INT((7+1)/2)= 4  
VE=INT((GVE+2)/3) = INT((5+2)/3)= 2

Het punt kan nu geplotted worden:

```
CURSOR TO (4,2)
BL=QC(4,2) OR QM(1,2)
PRINT BL;
```

Hierin is QC een array die het grafische scherm bevat. QM is de basismatrix, zoals gedefinieerd in fig. 3 en fig. 5. Plotten we een punt op een leeg scherm dan wordt

QC(4,2)=QM(1,2)=4. Maar stel dat we het volgende al op het scherm hadden staan:

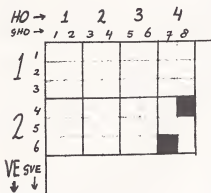


Fig. 7

Pixel toevoeging

waarde van die coördinaat kan ieder willekeurig beeldpunt geprint worden met deze methode. Daar de graphics in de VDUprom iets anders opgeslagen zijn als de standaard viditelcodes, dient men nog de volgende conversies uit te voeren:

- Als de graphiccode  $\rightarrow$  \$1F dan is de graphic te vinden op positie 255 - (berekende graphiccode OR 32)
- Als de graphic een van de vier uitzonderingsgraphics is (FF, CR, LF en ESC) dan moet die code omgezet worden naar de bijbehorende code. Een oplossing hiervoor is gegeven in LIST 1. De basicregels 750...770 presenteren dezelfde methode in List 2.

### Plotroutine in basic

In listing 2 staat een plotprogramma in basic. Het array QC bevat het grafische scherm. Het array QM is de basismatrix. Iedere coördinaat correspondeert met de code van een pixel. Subroutine 600 vult het array met nullen, en veegt zo het grafische scherm schoon. Na regel 1000 staan een paar demonstratieprogramma's:

1010: Plot een beeldpunt op positie 10,10.  
 1500: Trek een horizontale lijn op GVE=36 van GHO=1 tot 150.  
 1600: Teken een sinus.  
 1700: Wis de lijn die geplot is met regel 1500-1530

De regelnummers en variabelenamen zijn zodanig gekozen, dat ze geschikt zijn voor opname in de basicode-3 subroutines. Wil men deze plotroutine gebruiken voor basicode-3 dan moet de invoer van de coördinaten wel plaatsvinden in een percentage 0..1 (dat is 0..100%) Men dient er rekening mee te houden dat een plotroutine in basic met een grafisch array erg traag werkt. Als men een snel grafisch scherm wil hebben, moeten de volgende aanpassingen gepleegd worden:

- Omzetten van basic naar machinecode,
- In plaats van een grafisch array (QC), direct het videogeheugen gebruiken.

Om het grafisch scherm enigszins te normaliseren ligt het voor de hand om de plotcommando's conform het basicode-3 protocol te laten verlopen. Wie durft...

Het basicprogramma staat op het KIM BBS in de basicarea (15) onder de naam PLOT.demo.

Er is al een punt geplot op (GHO=7, GVE=6) en een op (GHO=8, GVE=4). Uitgedrukt in tekstcoördinaten staat er op (HO=4, VE=2) de blockgraphic \$12 geprint. Opnieuw gaan we nu een punt plotten op GHO=7, GVE=5.

```
CURSOR TO (4,2)
BL=QC(4,2) OR QM(1,2)
PRINT BL;
```

QC(4,2) = \$12 0001 0010

QM(1,2) = \$04 0000 0100

----- OR

BL = \$16 0001 0110 , en dat is

Er ontstaat graphic \$16. Het klopt ! Door altijd te ORren (tekenen) of te ANDen met het complement (uitwissen) met de huidige

```

D65COR ;Special char. dec. dos 65
        ;Converts graphic characters
        ;To the position in chargen. rom
        LDA    AHOLD
        BNE    1.F ;Special char decoder
        LDA    #$20 ;00:=space
1      CMP    #$0A
        BNE    2.F
        LDA    #0 ;0a:=0
2      CMP    #$0C
        BNE    3.F
        LDA    #$80 ;0c:=80 ex/
        JMP    XQCOR
3      CMP    #$0D
        BNE    4.F
        LDA    #$7F ;0d:=7f ex/
        JMP    XQCOR
4      CMP    #$1B
        BNE    5.F
        LDA    #$FF ;1b:=ff ex/
        JMP    XQCOR
5      CMP    #$7F
        BNE    6.F
        LDA    #$A0 ;7f:=a0
6      CMP    #$FF
        BNE    7.F
        LDA    #$20 ;Ff:=20
7      CMP    #$80
        BNE    XQCOR
        LDA    #$A0 ;80:=a0
XQCOR   STA    AHOLD
        RTS
    
```

### List 1: Conversie naar positie in vdu-rom

```

REM Plotroutine voor grafisch scherm in Dos-65 basic V2.00
REM Er moet een viditelkaraktergeneratoreprom op de vdukaart zitten.

REM initialiseer videoschermarray, definieer blockgraphicarray
20 DIM QC(80,24), QM(2,3):ST=1/96: GR=1/72
21 QM(1,1)=1: QM(2,1)=2: QM(1,2)=4: QM(2,2)=8: QM(1,3)=16: QM(2,3)=64
22 HO=79:VE=23
30 GOTO 1000

REM Zet de grafische mode uit
101 PRINTCHR$(27);:PRINT"G";:RETURN

600 REM zet grafisch scherm aan, en wis het
601 PRINT"Even geduld..."
602 FOR TI=1 TO 80
605 FOR PI=1 TO 24
606 QC(TI,PI)=0:NEXTPI:NEXTTI
608 PRINTCHR$(12);

REM Zet de grafische mode aan
612 PRINTCHR$(27);:PRINT"F";
615 RETURN
    
```

*Viditel*  
*Viditel*

```

REM Plot een punt op positie AT,AS ==(HO,VE)
REM Als CN=0 scherpunt word aangezet (voorgroondkleur)
REM Als CN=1 word het scherpunt uitgezet (achtergrondkleur)
620 GOSUB 700: GOTO 705

REM Kijk of de coördinaten binnen het scherm vallen
700 IF AT>=160 THEN AT=160
701 IF AS>=72 THEN AS=72
702 IF AT<1 THEN AT=1
703 IF AS<1 THEN AS=1
704 RETURN

705 REM De plotroutine himself...
706 TI=INT((AT+1)/2): PI=INT((AS+2)/3)
710 IF AT<=2 THEN 720 :REM zet grafische coörd. om naar teksteenheid
715 AT=AT-2: GOTO 710
720 IF AS<=3 THEN 730
725 AS=AS-3: GOTO 720
730 IF PI>24 THEN PI=24 :REM Grootst mogelijke waarden HO,VE
731 IF TI>80 THEN TI=80
740 IF CN=0 THEN QC(TI,PI)=QC(TI,PI) OR QM(AT,AS) :REM Plot een punt
745 IF CN=1 THEN QC(TI,PI)=QC(TI,PI) AND (NOT QM(AT,AS)) :REM veeg een punt
748 CURSOR TO PI,TI
749 GOSUB 612
750 IF QC(TI,PI)=0 THEN PRINT " ";:GOTO 101:REM conversie naar exception graph
751 IF QC(TI,PI)=10 THEN PRINTCHR$(64);:GOTO 101
752 IF QC(TI,PI)=12 THEN PRINTCHR$(&0080);:GOTO 101
753 IF QC(TI,PI)=13 THEN PRINTCHR$(&007F);:GOTO 101
755 IF QC(TI,PI)=27 THEN PRINTCHR$(&00FF);:GOTO 101
758 IF QC(TI,PI)>(&001F) THEN 762
760 PRINT CHR$(QC(TI,PI)+ASC("A")-1);:GOTO 101
762 AS=(QC(TI,PI)OR32)
764 IF AS=127 OR AS=128 THEN PRINTCHR$(&00A0);:GOTO 101
766 IF AS=255 THEN PRINT " ";:GOTO 101
768 AS=255-AS
770 GOSUB 101:PRINTCHR$(AS);:GOTO 612

REM Een paar demootjes:
1000 GOSUB 600 :REM wis grafisch scherm
REM Demo plot een punt
1010 CN=0:AT=10:AS=10:GOSUB 620:REM plot een punt op locatie 10,10

REM Demo trek een horizontale lijn
1500 FOR I=1 TO 150
1510 CN=0:AS=36:AT=I:PRINTI
1520 GOSUB 620
1530 NEXT

REM Demo teken een sinusgolf
1600 FOR I=1 TO 150
1610 CN=0:AT=I:AS=(SIN(I/150*6.28)*72)*0.5
1612 AS=AS+36 :REM Neem het midden v/h scherm als oorsprong
1620 GOSUB 620 :REM Plot de berekende punten
1630 NEXT

REM Demo wis de eerder getekende lijn uit
1700 FOR I=1 TO 150
1710 CN=1:AS=36:AT=I
1720 GOSUB 620
1730 NEXT
1800 PRINT "End demo"
1810 END

```

### Karaktergenerator versies

Voor de elektuur VDU-kaart zijn verschillende viditelkaraktergeneratoren in omloop. De verschillen zijn miniem:

- De octopus karaktergenerator (zie 6502kenner 47, art. Coen Boltjes) Deze karaktergenerator bevat de eerste 32 blockgraphics van \$00 tot \$20. Vanaf \$20..\$7F staat de octopus ASCII karakterset. De locaties \$80..\$FF bestaat uit dezelfde data, maar nu in inverse video. De opbouw van de basismatrix is 8x9.
- Viditel chargen V1.0. Dit is een gemodificeerde dos-65 karaktergenerator. In plaats van de controlcodes, zijn nu de viditel blockgraphics opgenomen. Er zijn vier uitzonderingen:
  - De CR controlcode,
  - de LF controlcode,
  - de FF controlcode,
  - de ESC controlcode.

Het is erg handig om deze graphics zichtbaar te maken als men met ED werkt, vandaar dat deze gehandhaaft zijn. De oorspronkelijk viditelcodes staan op een andere plaats. (zie deel 1) Het matrixformaat voor de graphics is 8x9

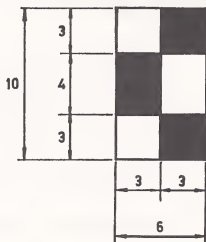
- Viditel chargen V2.0

Als V1.0, maar nu met een mini-matrix van 8x10. Het nadeel van V1.0 is dat de descenders van de g, j en p weg- vallen. Bij V2.0 is dit niet meer het geval. De indeling van de matrix is 3-3-4.

- Viditel chargen V3.0. Als V2.0, maar met minimatrix 3-4-3, geheel volgens de PTT-norm.

Viditel chargen. V2.0 is de meest universele. Door de CRTC een scanline minder te programmeren fungeert de karaktermatrix als 3-3-3 matrix. Dit heeft voordelen bij toepassingen voor een grafisch scherm. Bovendien valt het nauwelijks op, of men het middelste beeldpunt vetter ziet dan het onderste punt.

Opbouw van de mini-matrix in 6 x 10 matrix



Deel 1 is gepubliceerd in 6502 kenner nr. 54

### SCRED 5.2

Er is een opvolger voor Scred 5.0 (Basic-editor) De nieuwe versie 5.2 is alleen noodzakelijk indien men I/O-65 V2.11 gebruikt. Scred 5.2 is onafhankelijk van I/O-65 gemaakt. Bovendien is er nog een klein foutje uitgehaald. Bij de voorgaande versies was het niet mogelijk een karakter te inserten als de videopointer op de laatste regel stond. Nu kan dat wel.

### VIDITEL 2.2

Ook viditel-65 V2.2 gebruikt een routine uit I/O65, die niet in de jumtable staat. Bij gebruik van I/O65 V2.11 betekend dit, dat de CRTC-6845 niet correct geprogrammeerd staat na het verlaten van viditel, of bij het omschakelen van terminal- naar mainmode. Inmiddels is de ontwikkeling van de nieuwe viditelversie al zover gevordert, dat het geen zin meer heeft de oude versie aan te passen. Viditel 3.0 word ook onafhankelijk van I/O65. Meer over viditel in een volgend nummer.

DOS65K

This article is intended for those familiar with DOS65. For information about this english low-threshold-operation-system or english translation of article contact author.

Author: Erik van den Broek  
1e Atjehstraat 164  
1094 KX Amsterdam  
Netherlands  
tel 020 923237

Amsterdam, 10-3-'88

### INLEIDING

Dit artikel is nog niet 'af': op diverse punten zijn de aanpassingen van de E.C.65K aan de DOS65-configuratie nog niet geoptimaliseerd of getest: ACIA,RTC Deze optimalisaties zijn echter slechts 'het er uit halen wat er in zit' en niet noodzakelijk om DOS65 op de 65816 te laten draaien. Voor het verder verbeteren van het ontwerp, m.n. het verhogen van de klokfrequentie tot 4MHz zijn experimenten met een goede scoop en het verder uitwerken van de tijdsdiagrammen (zie literatuuropgave) noodzakelijk. Daar ik het nuttiger vindt om eerst software voor de 65816 te produceren, heb ik dit artikel nu reeds ter redactie van de 6502-kenner gebracht.

DOS65 V2.01 kan op diverse configuraties geïnstalleerd worden: JUNIOR, OCTOPUS,CPU-VDU, EC65K. Deze laatste, beschreven in Elektuur Computing 4, is de mooiste, vanwege de volgende voordelen:

- 1) Snel: 4Mhz klok \*
- 2) 16 Mbyte adresbereik, 16 bits, 8 en 16 bits accu en index-registers, 256 instructies, 24 adresserings-mogelijkheden, 'zero-page' en stack overal in onderste 65K
- 3) DMA en co-processing faciliteiten

\* Optioneel in deze beschrijving: hier 2 MHz.

Dit artikel beschrijft de wijzigingen, die in het originele ontwerp dienen te worden aangebracht, opdat DOS65 kan worden geïmplementeerd. Verder worden enkele geconstateerde fouten in het ontwerp behandeld. Als resultaat ontstaat een systeem, dat alle oude software in de 6502-emulatie-mode kan verwerken en 'uitstapjes' naar de 65816-native-mode toelaat, om met 16 bits en 'hogere' geheugenbanken te werken. In een volgend artikel worden macro's besproken, die 65816-objectcode generen, zodat men ook de oude assembler kan blijven gebruiken.

Noodzakelijke literatuur: Elektuur Computing 4  
overige raadplegingen: E.C. 5 (noodzakelijk bij 256k SRAM-kaart)  
E.C. 1 (opzetprint-interface van CPU)  
65816/65802 Assembly Language Programming  
door Michael Fischer  
Osborne / McGraw-Hill ISBN 0-07-881235-6  
(sterk aanbevolen, sterk duur (f 100,-))

### A) De CPU-kaart (E.C.4 pag 29)

Deze kan in principe geheel ongewijzigd blijven, doch bevat van zichzelf een onvolkomenheid: de schakeling rond N31, welke een NAND imiteert, deugt niet: door de relatief grote stromen door D1 en D2 ontstaat soms een te grote spanningsval over deze dioden, waardoor het DIR-sigitaal naar IC8 niet goed gedefinieerd is. Oplossingen: 1) vervang D1,D2,R15 en N31 door een poort van een (extra) 74F00 of 2) laat N31 het werk van b.v. N23 doen en gebruik de vrij-

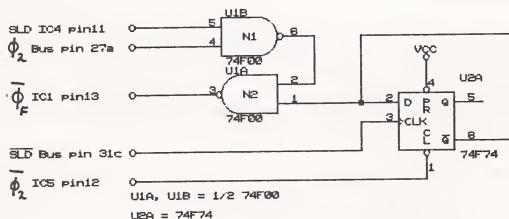
gekomen poort voor de NAND-functie.

De I/O65 EPROM wordt geleverd in een halfbenutte 2764, terwijl op de kaart slechts plaats is voor een 2732. Men kan zelf een 2732 programmeren, of IC 19 met een tweede voet ophogen, de poten 1,2,27 en 28 boven IC 18 laten bengelen en verbinden met +5V.

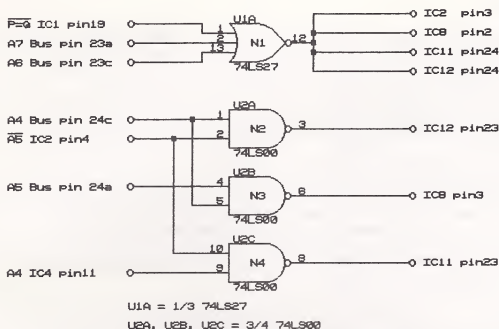
De CPU voorziet niet in de 1k garbage-RAM, welke I/O65 nodig heeft op het bereik 00E400 - 00E7FF. Het eenvoudigst installeert men een half benutte 6116 (dit bereik is 'slowed - down', dus de 6116 mag traag zijn) op een aparte print op de bus. Zie ook punt D5) verderop.

DOS65 gaat uit van het gebruik van via's en de daarin aanwezige synchrone timers en schuifregister. Het kloksignaal naar deze via's moet dus synchroon met de processorklok (en dus de 'valid-address-strobe') lopen als de via's geadresseerd worden en altijd 1MHz (of altijd 2MHz) zijn. Men kan dus niet zomaar fase-F gebruiken, omdat dan niet aan de synchronisatie-voorwaarde is voldaan. De volgende schakeling deelt fase-2 door twee als er geen slow-down is en laat fase-2 ongemoeid tijdens s.d. Haal pin 13 van N6 uit de voet en verbindt met de schakeling. De flipflop moet snel zijn en kan dus niet 1/2 IC 12 zijn; de NANDS zijn wellicht in de bij oplossing A1 ingezette 74F00 te vinden.

Het zo ontstane fase-F signaal, wordt ook gebruik als kloksignaal voor de VDU, FDC, VIA'S en ACIA in de 4MHz versie. Komt er geen 4MHz op fase-2 voor, dan hoeven alleen de via's van een synchrone 1MHz (of 2MHz) klok te worden voorzien. Bij gebruik van een grafische kaart kan pas optimale snelheid worden verkregen door een 4/2/1MHz versie te bouwen. Dit kan eenvoudig, door het signaal op pin 6 van N8 met een deler zoals de hier getekende fase-2-deler, nogmaals door twee te delen als de grafische kaart geadresseerd wordt.



FIGUUR I (fase-f aanpassing)



FIGUUR 2 (ADRESDECODERTJE)

N.B.: Alle ic's op deze kaart, dus ook IC1 mogen LS-typen zijn.

### B) De PSIO-RTC-kaart:

Voor minimaal functioneren van DOS65 is slechts een via (of zelfs pia) nodig voor I/O (toetsenbord). Men heeft dan echter geen modem, printer, etc., aansluiting. Beter is dan ook, de opzetprint van de CPU uit E.C.1, met uitbreiding van een konnektor en het verderop beschreven decodertje, te gebruiken (Zie voor de klok-ingangen van de VIA's onder A; er moet wellicht ook nog een 'page=\$E0-sigitaal' en een 'bank=0-sigitaal' gefabriceert worden indien ze niet reeds bestaan). Deze methode is niet getest, maar is op de RTC na identiek aan de PSIO-RTC-kaart. Men kan ook (zoals ik) een gewijzigde PSIO-RTC-kaart bouwen Wijzigingen:

Haal van IC 8 pin 2 EN PIN 3 uit de voet en verbindt met decoder; idem IC 2 pin 3, haal van IC 2 pin 2 uit de voet en verbindt met IC 4 pin 12/13. Plaats de jumpers A9,A10,A11,A12. Monteer voor IC 11 en IC 12 VIA's 65(C)22 i.p.v. PIA's. Laat de volgende pinnen naar buiten hangen:

21,22,23,24,25,35,36,37,38. Verbindt ze nu als volgt:

- 21 <--> IRQ (=bus 12a)
- 22 <--> R/W (=IC 4 pin 6)
- 23 <--> adresdecoder (zie verderop)
- 24 <--> adresdecoder (zie verderop)
- 25 <--> fase-F (=bus 28a) (zie onder A)
- 35 <--> A3 (=bus 25a)
- 36 <--> A2 (=bus 25c)
- 36 <--> A1 (=bus 26a)
- 38 <--> A0 (=bus 26c)

Door de klok-ingangen van deze VIA's, zoals hier aangegeven, met de aangepaste - zie onder A) EN ONDER D5) - fase-F i.p.v. fase-2 te verbinden, zijn de timers en schuifregisters ook bij slow-down weer goed te gebruiken.

### C) DE 256K SRAM-KAART

Om met een klokfrequentie van 4Mhz te werken, moet er snelle SRAM gebruikt worden, hetgeen op diverse kaarten kan, mits snelle poorten gebruikt worden (F,S of AS typen). Het kan o.a. met de 256K SRAM-kaart uit E.C.5, hoewel deze kaart niet zondermeer bij een 6502, OF ZELFS DE 65816 IN EMULATIE-MODE gebruikt kan worden, omdat hij op de databus ook het gemultiplexte banksignaal verwacht. Na een reset (waarna de 65816 automatisch emuleert) moet dus eerst een registerinhoud naar IC2 geschreven worden, voordat de kaart gedefinieerd is. Oplossingen:

- 1) Gebruik de kaart met 6264 SRAM's, verwijder IC 2 en IC 3, verbindt van voet van IC 3 pin 1 met pin 19 en leg de jumpers g,h,i,j,q,s,t,u,w,y. De kaart ligt nu altijd in het onderste 64k bereik. Zie ook verder onder punten D)2) en D)3)
- 2) Gebruik een andere (snelle) kaart voor bank 0 (die ook alleen maar geadresseerd wordt als er in die bank gewerkt wordt) en de 256K kaart elders (niet getest).
- 3) Schakel bij emulatie (buspin 27c is dan laag), middels een 74F157, over op wijziging C1 (niet getest).

### D) OVERIGE OPMERKINGEN

- 1) Op geen enkele kaart mogen de busaansluitingen 16a en 16c met massa verbonden zijn. Dit zijn namelijk (de enige) niet compatibele pennen t.o.v. de oude elektuurbus.
- 2) Voor alle in deze en Elektuur bouwbeschrijvingen genoemde F-typen kan men ook S- of AS-typen gebruiken. Deze zijn nog iets sneller, maar gebruiken (vooral S) meer stroom: F:0.8 mA/poort, AS: 1.5 mA/poort, S: 4 mA/poort.
- 3) Ter voorkoming van beeldruis kan men de CPU synchroniseren met de VDU. Zet dan het 16MHz signaal van de VDU op fase-H (=buspin 30c) en deel tot 8 MHz m.b.v. CPU-kaart IC 12: haal pin 6 en pin 8 van IC 1 uit de voet en verbindt van IC 12 pin 8 met pin 12, pin 11 met fase-H (=buspin 30c) en pin 9 met pin 1 van IC 2.
- 4) Verbindt de huisjes van kristallen altijd met massa (zonder ze te verhit-ten), alleen dan werken ze betrouwbaar.
- 5) Met slow-down op het gehele adresbereik 00E000-00FFFF (b.v. door jumper u op de 256K SRAM-kaart) wordt wat betreft snelheid niet het onderste uit de kan gehaald. Met name de beeldschermroutines van de I/O EPROM zorgen er voor, dat indien men, b.v. tijdens assemblage, op het scherm wil volgen wat er gebeurt, de tijdwinst gering is. Een eenvoudig experiment (dat bij iedereen anders kan uitpakken) met selektievere vertraging, kan hier wonderen doen. Bij mij bleek bij 2MHz in het geheel geen slow-down nodig, ook niet bij gebruik van n-mos 6522. De definitieve kloksnelheid zal dus 4/2MHz kunnen worden, daarover later.

## 053-303902

### Hét 6502 Info Board

### Bespiegelingen 65(C)02-'88++

Als je, zoals ik, lid bent van een computerclub lijkt je na een aantal jaren vast te lopen. Weet je ook van te voren. Geen man overboord, maar ook niet echt leuk. (En van neveneffecten, die hierdoor onwillekeurig groeien in zo'n sfeer is nogal wat te merken geweest binnen deze club. Ook niet echt bouwend aan het geheel.)

In computerland gaan alle dingen zo snel, dat zelfs profs moeite hebben het tempo bij te houden. Tegen de tijd, dat je enigermate vertrouwd begint te raken met jouw systeem komt er weer een nieuwe klasse hard- en software. Heb je dan nog iets aan jouw club, of moet je er maar mee ophouden?

Dat zou het langzaam doodbloeden van die gebruikersclub kunnen impliceren. Ook geen wereldschokkend nieuws, maar is dat nou nodig? Zouden er niet slimmere dingen met zo'n club mogelijk zijn, dan hem na een aantal jaren naar de vergetelheid te verwijzen? (Onder het b.v. vreemd schrijven over 'gladde Dos-65'-jongens en opmerkelijke teksten over andere programmeertalen.)

Een processor is met zo'n tien jaar verouderd, het systeem, waar mee gewerkt wordt blijkt plotseling tien keer kleiner dan erg goedkope fabrieksmachines (waar op een 'kwade' dag iedereen mee schijnt te werken, voornamelijk IBM, als wij de HCC mogen geloven). De performance van de club- software is op zijn minst beduidend krakkemikkiger, dan voordelige pakketten voor eerder genoemde fabrieks-hardware. Aansluiten op een standaard is moeilijk. Gerommel binnen de club met hard- en software maken jouw eigen voortgang er ook niet beter op enz.

Op het moment, dat een dergelijk gevoel je besluip (wat heet, over je heen walst, als je bij voorbeeld eens in 'n HCC-aflievering kijkt) dan moet je z'n en onz'n van elkaar gaan scheiden. Dat gevoel krijg je m.i. n.l. door problemen met de huidige spullen en daar tegenover verleidelijke reclame uit computergebieden, waar we eigenlijk geen of nauwelijks belangen hebben. (Wellicht eventueel later). Of het moet zijn dat je echt een grote machine nodig hebt bv. voor studie. (Fortran, Modula, Pascal enz.) Of als een computer iets anders voor je is, dan een leuk hard- of/ en software-experimenteerterrein.

Derhalve, vervolgens het volgende.

Waarom wordt er gecomputerd?

Wel, omdat het moet in een bedrijf. (Voor hogere efficiency en zo...) Dat vereist gestandariseerde hard- en software van ruime omvang, met snelle CPU's en grote back-up liefst gekoppeld aan netwerken (en multitasking natuurlijk). Begrip van het onderliggende systeem op machineniveau is niet nodig en ook niet interessant. De hobbyist zal hier over het algemeen weinig aan hebben. Het terrein van de informaticus, de bedrijfsmatige gebruiker etc.

Dan is er de video-sector, de medische sector, de CAD-CAM etc.-sector. Daar moet men ontwikkelen met geavanceerde digi-video-systemen. Minstens even zware eisen als hiervoor. Ook hier ligt de nadruk op applicatie. Te duur voor de hobbyist en nauwelijks zinnig toe te passen in de prive-sfeer.

Vervolgens de half prof- half hobby-sector. Deze mensen kunnen het beste een apparaat kopen, dat zoveel mogelijk ruimte biedt voor alle experimenten naast profgebruik. Hardware-bouw is nihil, en in het gunstigste geval programmeren men in een of meer talen en draait verder applicatiesoftware. (Bv. '88 Atari ST's, Amiga en Schneider). In meer of mindere mate kan er bij deze gebruikers echt noodzaak zijn, te werken met een dergelijke ruime "environment". Maar veelal zijn er ook andere redenen. Sommige gebruikers uit deze categorie zijn onderhevig aan wat ik zou willen noemen "mode-invloeden". Bij nieuwe spectaculaire ontwikkelingen wordt de computer snel tweede hands verkocht of gaat de kast in en spoorsslags wordt het nieuwe spektakel in huis gehaald. Of deze computeraar in staat is geweest de vorige machine tot op de bodem aan te spreken? Vrijwel nooit. Laat staan 1 Meg met 200k operating system.. Beetje onrustig. Arme 64k machines.. Vaak weet men overigens vaak wel uitstekend, hoe men zijn vrienden en neefjes met het nieuwe moet imponeren. Nou ja... als dat dan gelukkig maakt, vooruit dan maar.

Wel interessant materiaal voor een hobbyist of student, die zich serieus met grote compilers of interpreters wil bemoeien. Minder voor diegenen, die de computer hard- en software-matig naar een echt eigen configuratie willen laten groeien.

Verder maar met de professionele programmeur/analist. Deze systeem/applicatie-ontwikkelaar moet juist wel een behoorlijk begrip van onderliggende processen hebben. En de overkoepelende bedrijfsvormen. Met

hardware bemoeit hij zich matig. Hij denkt in grote hardwareblokken. Deze materie is omvangrijk. Hij gebruikt liefst gereedschappen, die zoveel mogelijk machine-onafhankelijk zijn (C, Fortran, Assembler, tools). Toch ook wel interessant voor de hobbyist. Het zou aardig zijn als de hobbyist met eventueel minder omvangrijke compilers (of interpreters) een kijkje zou kunnen nemen in dat keukentje. Dus kleinere versies van prof-gereedschappen zijn wel degelijk van belang binnen computerclubs.

Dat is dan duidelijk. Waar gaat het nu eigenlijk om.

Om ons. Voor de 6502-Kenners zijn er eigenlijk geen grenzen aan te geven. Zelfbouw kan. Je kunt ook een grote moderne machine kopen. Professionele toepassingen zijn er in ons geval vrijwel niet. Het is dus de vraag of IBM etc. voor de hobby zinnig is. Hobbymatig is er echt weinig aan te beleven. Of het moet voor studie van programmeertalen zijn of zoiets.

Goed. Stel ik blijf het doen met mijn 64k-bakkie. Hard- en software-ondersteuning binnen de club moet dan wel optimaal zijn!!, anders gebeurt het volgende.

Een goede Hobbyist snuffelt niet zo lang, tot hij automatisch aan een andere computercluster raakt. Zo heb ik het ervaren. Als hij het niet bij de eigen club vindt, stapt hij op. Voor je het weet wordt een Atari of een kloon gekocht. En hoewel het niet nodig is, verlaat hij zijn oude club vrij snel en sluit zich aan bij een nieuwe.

Ik kocht naast mijn Ohio en Dos-65 1.0 -> 2.0 systemen een 1Meg Atari met veel talen, literatuur enz. Geen onvolledige interpreters, alle talen beschikbaar, ruim, volledig en nu maar programmeren. Of er een wereld voor je opengaat. Wat heet! Motivatie daartoe: onvrede. (En in mijn geval studieneiging.) Als de ondersteuning beter zou zijn geweest binnen deze club, dan had ik nog gewacht met een dergelijke aankoop. Ondertussen ben ik toch nog maar lid gebleven. Maar voor een heel groot deel uit een ouderwets soort trouw, er zitten aardige mensen in en misschien gaat het nog eens beter lopen binnen de club.

Had allemaal niets te maken met technologie-ontwikkelingen. Ook niet met hysterisch geschreeuw in HCC-artikelen en reclame!

IBM-gebruikers hebben andere noden als ik. Een duur CAD-CAM-systeem hoef ik niet. Maar een onaf Basic of Forth, dat op Dos 65 1.0 draaide en niet op 2.0, maakt je ziek. Geen Tiny-Pascal voor 2.0 of zoiets is onzin. En struikelen over de bugs in Ohio-applicaties (vroeger) geeft ellende. De oude programma's allemaal moeten converteren, omdat 'n oude programmeertaal niet meer op een 65(C)02 hardwaregemodificeerd systeem wil draaien, terwijl je die programmatuur dagelijks moet gebruiken, is bijna een ramp. En een niet af Basic is zelig. En dan nog gebruikers-onvriendelijkheid in programma's. Dat alles is tegengesteld aan de regel, dat de computer uiteindelijk het werk moet doen bij applicatie-gebruik en niet de gebruiker! En elke systeemsoftware, die niet in orde is zelf op machineniveau in orde maken, is belachelijk of nauwelijks te doen en we hebben wel andere besognes. Hoe leuk ook dat zelf ontwikkelen.

Dus HCC goes IBM? So what! 30k vrij te klein? Nou, probeer zoiets maar eens zinnig vol te programmeren. 65(C)02 weg-gooien? Welnee, meestal ruim voldoende voor de job, die jij er mee wilt doen. Als je maar slim programmeert! In de industrie wordt het merendeel van de besturingsprocessen met dit soort processors en weinig geheugen gedaan. Zin en onzin, daar gaat het hier dus om. Wat let je, toch ook 's iets anders te proberen. En om lekker met hardware te rommelen. Daarop een goed artikel in het blad te plaatsen. En goede, gestructureerde, afgeronde, gebruikers-vriendelijke software te ontwikkelen, of dat nou op een 6502, op een 68020 of op een "tachtiger" gebeurt, als het echt kwaliteit heeft, geeft dat veel bevrediging. En goed lopen doet het dan ook zeker. Een slim geprogrammeerde kleine processor als bv. de 65(C)02 heeft een prima performance.

Maar met onvolledig gereedschap houdt alle enthousiastie op. Of ombouw van een systeem en dan merken, dat veel programma's moeten worden geconverteerd, omdat de software niet compatibel is.... Op zo'n moment zou ik even wachten met openbaar maken, tot dat wel in orde is.

Naar mijn mening zou het ook goed zijn, als de 6502 Kenners een andere naam zouden gaan voeren (Hard- en Software-Club Nederland of zie maar).

Naast dat er een aantal systemen standaard-support krijgen, moet er ook ruimte voor andere systemen zijn. Wellicht komt er 'n systeem bovendrijven, dat ook die

standaardsupport verdient.

(Nauw) contact realiseren met andere clubs lijkt mij ook best interessant. Beetje bundelen van intellect?

Wellicht zou een landelijk verband gestart kunnen worden, nu de HCC IBM en ex-hobby is gegaan.

Een Node hoort er bij. (En dus ook weer ondersteuning!)

En ook contact met een publicerend landelijk blad, zoals vroeger enigermate met Elektuur (nu wellicht met Radio Bulletin). Misschien eens contact met Hobbyscoop? Dat is goed voor toevloed van leden.

Software moet centraal en voor de leden goedkoop verkrijgbaar zijn, op disk voor alle systemen, als harcopy en eventueel op band of via telefoon. En leift ook hardware. Centraal beheerd en gecontroleerd.

Moet ook goed in elkaar zitten, vrijwel (liefst helemaal) zonder bugs en gebruikersvriendelijk, zonder dat zulks te verdragend gaat werken. Niet te vroeg uitbrengen, anders gaan dergelijke versies een eigen leven leiden... En ga niet rommelen met systemen als echt volledige ondersteuning nog niet is te realiseren.

En om op het begin van dit epistel terug te komen, als het lekker gaat binnen deze club, laat je je niet zo snel meer onrustig maken door voor de hobbyist niet al te belangrijke agressieve reclame, als zou iedereen een kloon moeten kopen om nog zinnig te kunnen computeren..

Ik denk, dat een club als deze belangrijk is en dat kan zo blijven. Liefst zou ik de importantie zelfs wat zien groeien. Maar dan moet er orde op zaken zijn.

Wat ik hier heb opgeschreven is een poging tot iets positiefs. Het is vast niet volledig.

Ik houd mij aanbevolen voor op-, aanmerkingen en aanvullingen.

Gerard J. Reitsma  
Schouw 19  
1261 LE Blaricum.  
Tel 02152-56645.

### REACTIE.

Het grote voordeel van het feit dat je de redactie van een blad doet, is dat je eventueel als eerste op een stuk kunt reageren. Ik ben van mening dat het stuk van Gerard een aantal zeer zinvolle opmerkingen bevat. Een deel van deze zaken zijn in het (recente) verleden ook al binnen het bestuur aan de orde geweest. Voorbeelden hiervan zijn een naamsverandering, samenwerking met andere verenigingen en contact met landelijke tijdschriften.

Ik wijs er nadrukkelijk op, dat het volgende mijn persoonlijke mening weergeeft en niet opgevat moet worden als het standpunt van het bestuur.

In principe heeft Gerard op bijna alle punten gelijk. Natuurlijk moeten we proberen de systemen binnen de club een professionele ondersteuning te bieden en trachten software te verspreiden die foutvrij, gebruikersvriendelijk en goed toepasbaar is. Hier komt echter het eerste probleem om de hoek kijken. Wij zijn een club van hobbyisten. Hobbyisten zijn per definitie enthousiast en zijn in de regel niet op een professionele manier met hun hobby bezig. Dat betekent dat iets uit de handen van een hobbyist in de ogen van de maker altijd foutvrij, gebruikersvriendelijk en goed toepasbaar is. Na verspreiding blijkt dan vaak dat er aan het product nog iets mankeert. Ik vind dat niet erg want meestal is er dan wel weer een andere hobbyist die in staat is het product aan zijn wensen aan te passen. Wat belangrijk is in dezen is een goed informatie-circuit om alle (toekomstige) gebruikers van het euvel en de gevonden oplossing op de hoogte te stellen. Ik denk dat wat dat betreft binnen de club mogelijkheden genoeg aanwezig zijn.

Een tweede punt waar ik op wil reageren is de verspreiding van de producten. Ik ben het met Gerard eens dat dit centraal en tegen kostprijs moet gebeuren. Ik ben daarom ook van plan dit vanuit mijn positie als bestuurslid en redacteur te trachten hiervoor een goed systeem op te zetten. Maar ook hier komt weer het punt naar voren dat we hobbyisten zijn. De collega's in het bestuur hebben allen een drukke baan en hebben aan hun bestuurstaak meestal al de handen vol. Degene die dit uitlegt als een verkapte oproep voor medewerkers ziet dat dan ook goed; we kunnen enige ondersteuning heel goed gebruiken en degene die denkt dat hij naast zijn dagelijkse werkzaamheden nog iets in organiserende zin voor de club kan doen zal met

open armen worden ontvangen.

Dan kom ik tenslotte tot het derde punt, de ondersteuning van de systemen. Van oorsprong zijn wij een club van mensen die zelf computers bouwen. Dit betekent dat de bouwsels op het gebied van mogelijkheden meestal lang niet kunnen concureren met de kant en klaar gekochte systemen. Ook een hard- en software ondersteuning zoals die door de diverse bedrijven met hun honderden werknemers gedaan worden is natuurlijk binnen onze club niet mogelijk. Dit betekent dat we niet in staat zijn tientallen talen aan te bieden of operating systemen volgens de laatste stand der techniek met windows en muizen. Wat krijgt u dan wel? Contact met een aantal enthousiaste mensen die tot en met het laatste weerstandje precies weten hoe hun systeem werkt en daar uren over kunnen (en vaak ook zullen) vertellen. Het merendeel van deze leden is erg actief en het komt voor dat deze mensen tot geniale oplossingen komen. Op bijeenkomsten van onze club liggen vaak de hete soldeerbouten naast de systemen en worden goed werkende systemen vaak gesloopt om te kijken of met de onderdelen een niet goed werkend systeem aan de praat te krijgen is. Ik denk dat de trotse bezitter van een gekocht systeem dergelijke dingen liever niet met zijn systeem uithaalt.

Op basis van het voorgaande wil ik de volgende conclusies trekken. Gerard heeft gelijk maar het feit dat we een club van hobbyisten zijn maakt dat een professionele ondersteuning en producten van een hoge kwaliteit niet altijd mogelijk zijn. Wel is het zo dat er voor een geconstateerd probleem altijd een oplossing gezocht en meestal ook nog gevonden wordt. De club bestaat uit knutselaars en zolang er knutselaars zijn heeft de club in mijn ogen een bestaansrecht. Wel is het belangrijk dat de knutselaar met zijn producten, problemen en opmerkingen van zijn zolderkamertje afkomt en contact zoekt met anderen. De club biedt hiervoor het blad, het bulletin board, de bijeenkomsten en een groep enthousiaste bestuursleden die je bijna bij nacht en ontij mag storen om over de hobby te praten. Ik zou daarom willen uitschreeuwen:

KOM VAN JE ZOLDERKAMERTJE EN LAAT WAT VAN JE HOREN!

Gert van Opbroek  
Bateweg 60  
2481 AN Woubrugge  
01729-8636

### VRAGENRUBRIEK.

Het is de bedoeling dat in deze rubriek vragen gesteld kunnen worden over alles wat met de club of de hobby te maken heeft. Iedereen die een vraag in deze rubriek gesteld wil hebben, kan de vraag opsturen naar de redactie. Indien de vraagsteller daar prijs op stelt, kan de vraag ook op het Bulletin Board gezet worden. Antwoorden op gestelde vragen kunnen gericht worden aan de redactie of direct aan de vraagsteller.

Aangezien het beleid van de redactie zodanig is, dat telefoonnummers slechts geplaatst worden na uitdrukkelijke toestemming van de eigenaar, bij het stellen van de vraag s.v.p. vermelden of het telefoonnummer vermeld mag worden.

De heer H. Zimmermann werkt met een Junior met elekterminal en cassette recorder. Hij wil graag dat het programma Ursuspator op het scherm aangeeft wanneer men schaak staat. Hij heeft hierover de volgende vragen:

In het schaakprogramma "Ursurpator" (zie boekje van H.G. Muller, blz. 39) is aangegeven dat door de variabele "00B9 EXTRA" (High Byte MSCHK) te zien is of men schaak staat. In de praktijk blijkt mij dat dit pas geschiedt door de zet waarmee het schaak opgeheven wordt.

Is er iemand die dit ook heeft vastgesteld en hoe is dit op te lossen?

Moeten er parameters van te voren ingebracht worden?

Hoe luiden de volledige namen van de met afkortingen aangegeven variabelen?

Reacties s.v.p. aan:

H. Zimmermann  
Woesterbergweg 9  
8166 HE Emst

### GEVRAAGD:

Parallel keyboard

E.F. v/d Broek  
1e Atjehstraat 164 III  
1094 KX Amsterdam  
020-923237

```
SCR # 140
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 ( )
2 ( Dit is hetzelfde programma als in 'C' gepubliceerd is in de )
3 ( 6502-Kenner nr 53. Deze versie is geschreven in 79-standaard )
4 ( Forth door: )
5 (     Gert van Opbroek )
6 (     Bateweg 60 )
7 (     2481 AN Woubrugge )
8 (     01729-8636 )
9 ( Copyright: KIM-gebruikersclub Nederland )
10 ( )
11 ( Het programma is geschreven op een JUNIOR met PROTON-SENIOR )
12 ( DOS. )
13 ( )
14 ( )
15 -->
```

```
SCR # 141
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 ( )
2 ( DAG_FACTOR bepaalt bij een ingevoerde datum een dagnummer. )
3 ( Door dit dagnummer modulo 7 te nemen, krijgen we de dag in )
4 ( week waarbij de zondag code 0 heeft. )
5 ( Het verschil tussen de dagnummers van twee data, geeft het )
6 ( verschil in dagen tussen deze twee data. )
7 ( Het nulpunt van de dagnummers ligt in 1985. )
8 ( )
9 : DAG_FACTOR          ( dag maand jaar -- dagfactor )
10 DUP >R OVER >R      ( Kopieer jaar en maand naar R )
11 1985 - 365 * SWAP 1- 31 * + SWAP 4 - +
12 R@ 3 <              ( Januari of februari ? )
13 IF R> DROP R> 1- DUP 4 / SWAP 100 / 1+ 3 * 4 / - +
14 ELSE R> 4 * 23 + 10 / - R> DUP 4 / SWAP 100 / 1+ 3 * 4 / - +
15 THEN ;              -->
```

```
SCR # 142
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 ( )
2 : MOD7 7 MOD 7 + 7 MOD ; ( Bepaal modulo 7 ook voor < 0 )
3 : MYSELF LATEST PFA CFA , ; IMMEDIATE          ( Recursie. )
4 ( )
5 ( DATUM_WEEKNR bepaalt bij de datum het weeknummer en de dag )
6 ( in de week. Hierbij telt de maandag als eerste dag in de )
7 ( en wordt het weeknummer bepaald door het aantal donderdagen )
8 ( in het betreffende jaar. )
9 ( )
10 : DATUM_WEEKNR      ( dag maand jaar -- weekdag -nummer )
11 DUP >R              ( onthoud het jaar op de return st. )
12 1 OVER 1 SWAP DAG_FACTOR ( Op welke dag viel 01/01 ? )
13 4 OVER MOD7 -        ( Correctie voor 01/01 op of )
14 0< IF 0 ELSE 1 THEN   ( na donderdag )
15 >R >R              --> ( Onthoud klad en correctie )
```

```

SCR # 143
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 (
2   DAG FACTOR DUP MOD7          ( De gevraagde datum; bepaal weekdag )
3   SWAP 7 / R> 7 / -          ( Het aantal weken vanaf 01/01 )
4   OVER 0= IF SWAP 7 + SWAP 1- THEN ( Corrigeer voor de zondag )
5   R> +          ( Sommeer de correctie )
6   DUP 53 = IF DROP          ( Beschouw uitkomst 53 apart )
7       1 1 R> 1+ DAG FACTOR MOD7
8       5 < IF 1 -          ( zo, ma, di, wo: week 1 )
9           ELSE 53          ( do, vr, za )
10          THEN
11          ELSE DUP          ( week 0: dezelfde week als 31 12 )
12          0= IF DROP 31 12 R> 1- MYSELF SWAP DROP
13              ELSE R> DROP
14              THEN THEN
15 ;
-->

```

```

SCR # 144
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 (
2 ( CONVERTEER DATUM converteert een datum in dd mm jj of dd mm )
3 ( jjjj naar d ww jjjj - formaat. Bovendien wordt gecontroleerd )
4 ( of de ingevoerde datum bestaat. 0 --> datum ongeldig 1 --> )
5 ( conversie gelukt. )
6
7 : CONVERTEER DATUM          ( dd ww jj[jj] -- d ww jjjj f )
8   DUP 100 < IF 1900 + THEN
9   >R >R >R          ( Onthoud jaar maand dag op R-stack )
10  R> R@ SWAP >R          ( Haal de maand van de R-stack )
11  DUP 1 < SWAP 12 > OR          ( Controleer de maand )
12  IF R> DROP R> DROP R> DROP 0          ( maand ongeldig )
13
14
15
-->

```

```

SCR # 145
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 (
2   ELSE R@ 1 < R@ 31 > OR R@ 30 > R> R@ SWAP >R DUP 2 =
3   OVER 4 = OR OVER 6 = OR OVER 9 = OR SWAP 11 = OR AND OR
4   R@ 28 > R> R@ SWAP >R 2 = AND
5   R> R> R@ OVER >R ROT DUP >R ROT ROT          ( Copieer d,m,j )
6   DUP >R          ( Jaar nogmaals op R-stack )
7   DAG FACTOR 1 3 R> DAG FACTOR < 0= AND OR
8   IF R> DROP R> DROP R> DROP 0          ( maand onjuist )
9   ELSE R> R> R@ OVER >R DATUM WEEKNR          ( bepaal weekdag en week )
10      DUP 50 > R@ 2 < AND -          ( week 52 of 53 van vorig jaar )
11      IF R> DROP R> 1-
12      ELSE DUP 2 < R> 11 > AND          ( week 1 van volgend jaar )
13      IF R> 1+
14      ELSE R> THEN THEN 1          ( converteren gelukt )
15  THEN THEN ;
-->

```

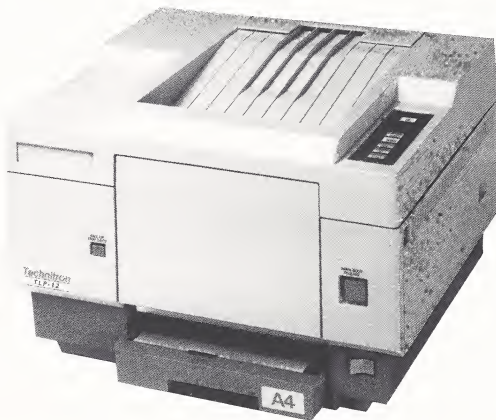
```
SCR # 146
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 ( )
2 ( LEES DATUM leest de geformateerde datum in en zet deze in )
3 ( het juiste formaat op de stack )
4
5 : LEES DATUM          (                               -- d m j )
6 ." Geef de datum dd/mm/jj of dd/mm/jjjj " PAD 64 EXPECT
7   PAD C@ 0= IF 0
8       ELSE PAD 1-    ( Breek de invoer op in drieën )
9       3 0 DO 0 SWAP 0 SWAP CONVERT SWAP DROP LOOP
10      DROP 1
11      THEN ;
12
13
14
15 -->
```

```
SCR # 147
0 ( ***** DATUM --> WEEKDAG CONVERSIE *****          GEVOP: 05/03/88 )
1 ( )
2 : DWJ BEGIN LEES DATUM WHILE CR CONVERTEER DATUM
3   IF ." De geconverteerde datum is :-"
4       0 <# # # # DROP DROP 45 HOLD 0 # # DROP DROP
5       45 HOLD DUP 0 # #> TYPE CR
6       ." De datum valt op een "
7       DUP 1 = IF ." maandag"      ELSE
8       DUP 2 = IF ." dinsdag"     ELSE
9       DUP 3 = IF ." woensdag"    ELSE
10      DUP 4 = IF ." donderdag"   ELSE
11      DUP 5 = IF ." vrijdag"     ELSE
12      DUP 6 = IF ." zaterdag"    ELSE
13      ." zondag" THEN THEN THEN THEN THEN THEN THEN DROP
14      ELSE ." Onjuiste datum" THEN CR
15      REPEAT ;
```

```
SCR # 148
0 ( NOOT VAN DE REDACTIE. )
1 ( ===== )
2 ( )
3 ( De redactie streeft ernaar alle gepubliceerde programma's )
4 ( ook op het Bulletin Board, 053-303902, beschikbaar te )
5 ( stellen. Deze F4th screens zullen daarom ook op het BBS )
6 ( staan. Nu doet zich het probleem voor dat niet bekend is )
7 ( wat het beste formaat is om ze op het BBS weg te schrijven )
8 ( zodat iedereen ze zonder al te grote problemen in kan lezen. )
9 ( )
10 ( Wie weet een manier om een tekstfile in F4th in te lezen en )
11 ( wie schrijft hier eens een programma voor. )
12 ( )
13 ( Reacties graag sturen naar de redactie of op het BBS t.a.v. )
14 ( de redactie van de 6502-Kenner of t.a.v. Gert van Opbroek. )
15 ( )
```

# TECHNITRON TLP-12 LASER PRINTER

## — U HEEFT EIGENLIJK GEEN ANDERE KEUZE!



- 12 pagina's per minuut (max.)
- tot 10.000 afdrukken per maand
- 8 ingebouwde lettertypes;  
32 afdruk-combinaties
- unieke "FontMaker" service
- unieke "FormsMaker",  
formulier- en logo service
- 3 ingebouwde hardware-  
emulaties
- flexibele in- en uitvoer van papier

**Technitron**  
**DATA**

Technitron Data B.V.  
Zwarteweg 110, Postbus 14,  
1430 AA Aalsmeer  
tel. 02977-22456  
telefax 02977-40968  
telex 13301

Vestigingen in:

BONDSREPUBLIEK DUITSLAND - DENEMARKEN - ENGELAND - FRANKRIJK - ITALIË - NOORWEGEN - VERENIGDE STATEN - ZWEDEN